



**DEPARTAMENTO DE INGENIERÍA TELEMÁTICA**

Escuela Politécnica Superior  
Universidad Carlos III de Madrid

PROYECTO FIN DE CARRERA

**Virtual CEIP: DISEÑO  
DE UN ENTORNO VIRTUAL  
DE APRENDIZAJE PARA UN  
COLEGIO DE EDUCACIÓN  
INFANTIL Y PRIMARIA**

Autora: **Selene Pinillos Franco**

Tutora:  
**Maria del Carmen Fernández Panadero.**

Noviembre 2010

El tribunal:

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto fin de carrera el día \_\_\_\_  
de \_\_\_\_\_ de \_\_\_\_ en Leganés, en la Escuela Politécnica Superior  
de la Universidad Carlos III de Madrid, acuerda otorgarle la calificación de:

EL PRESIDENTE

EL SECRETARIO

EL VOCAL

# AGRADECIMIENTOS

Llegar hasta aquí solo he podido conseguirlo con la ayuda de mis padres que han podido pagarme toda mi educación hasta el día de hoy. Finalizar este proyecto ha sido todo un reto para mí ya termina una etapa de mi vida y empieza otra porque ya puedo decir bien alto: SOY INGENIERA.

Quiero agradecer este proyecto también a mi novio que si no me mandase ponerme las pilas todos los días seguro que no lo hubiese acabado.

Se lo agradezco también a mi hermana, familia y amigos y, cómo no, a mi tutora que ha sufrido como yo para sacar este proyecto adelante.

# Resumen

Los colegios públicos han tenido dificultades en adaptarse a las nuevas tecnologías. Las aplicaciones educativas existentes en el mercado no siempre se adaptan a las necesidades del profesor y las que mejor pueden adaptarse tienen licencia.

Este proyecto intentará solventar una necesidad real de un colegio de primaria que dispone de un aula informática con una dotación informática del 2002.

Para ello se creará un mundo virtual que será una réplica del mundo real. Dentro de este mundo existirán tres aplicaciones que simularán las actividades que el alumno utiliza normalmente en el aula. El contenido de estas aplicaciones podrá ser manipulado por el profesor mediante archivos de configuración utilizando para ello archivos que normalmente sabe utilizar como documentos de texto, imágenes, audios o videos.

# Índice

<b>Resumen</b>	<b>ii</b>
<b>1 INTRODUCCIÓN</b>	<b>1</b>
<b>2 OBJETIVOS</b>	<b>4</b>
<b>3 ESTADO DEL ARTE</b>	<b>6</b>
3.1 CONCEPTOS GENERALES . . . . .	6
3.2 MUNDOS VIRTUALES . . . . .	7
3.3 WONDERLAND . . . . .	9
3.3.1 QUÉ ES WONDERLAND . . . . .	10
3.3.2 ARQUITECTURA . . . . .	10
3.4 UNITY3D . . . . .	13
3.5 XML . . . . .	14
<b>4 DISEÑO DE ESCENARIOS</b>	<b>15</b>
4.1 TOMA DE DATOS . . . . .	15
4.2 DISEÑO DEL EDIFICIO . . . . .	16
4.3 DISEÑO DEL AULA . . . . .	17
<b>5 APLICACIÓN PARA EL PROFESOR</b>	<b>20</b>
5.1 ANALISIS DE OPCIONES . . . . .	20
5.2 IMPLEMENTACIÓN . . . . .	23
5.3 DISEÑO DE JUEGOS Y ARCHIVOS DE CONFIGURACIÓN	24
5.4 EVALUACIÓN DEL JUEGO . . . . .	27
<b>6 MANUAL DEL PROFESOR</b>	<b>28</b>
6.1 CONTENIDO DE MATEMÁTICAS . . . . .	30
6.2 CONTENIDO DE LENGUA . . . . .	32
6.3 CONTENIDO DE CIENCIAS . . . . .	33
<b>7 APLICACIÓN PARA EL ALUMNO</b>	<b>34</b>

<b>8</b>	<b>MANUAL DEL ALUMNO</b>	<b>39</b>
8.1	APLICACIÓN DE MATEMÁTICAS . . . . .	42
8.2	APLICACIÓN DE LENGUA . . . . .	43
8.3	APLICACIÓN DE CIENCIAS . . . . .	44
<b>9</b>	<b>VALIDACIÓN E INTEGRACIÓN</b>	<b>46</b>
<b>10</b>	<b>PLAN DE TRABAJO Y PRESUPUESTO</b>	<b>49</b>
10.1	PLAN DE TRABAJO . . . . .	49
10.1.1	REQUISITOS INICIALES . . . . .	49
10.1.2	ESTADO DEL ARTE . . . . .	49
10.1.3	CREACIÓN DE MUNDOS 3D (SKETCHUP) Y CREACIÓN DE ESCENARIO EN WONDERLAND . . . . .	50
10.1.4	HERRAMIENTA DEL PROFESOR Y PRUEBAS . . . . .	50
10.1.5	CREACIÓN DE JUEGOS EN UNITY3D Y PRUEBAS . . . . .	50
10.1.6	PRUEBAS DE INTEGRACIÓN . . . . .	50
10.1.7	INTEGRACIÓN FINAL . . . . .	51
10.1.8	REDACCIÓN DE LA MEMORIA . . . . .	51
10.2	PRESUPUESTO . . . . .	51
10.2.1	DESGLOSE POR ACTIVIDADES . . . . .	51
10.2.2	SALARIOS POR CATEGORÍA . . . . .	52
10.2.3	GASTOS DE PERSONAL . . . . .	52
10.2.4	RECURSOS MATERIALES EMPLEADOS . . . . .	52
10.3	COSTES INDIRECTOS . . . . .	52
10.4	MARGEN DE BENEFICIO . . . . .	53
10.5	RESUMEN DEL PRESUPUESTO . . . . .	53
<b>11</b>	<b>CONCLUSIONES</b>	<b>54</b>
<b>12</b>	<b>TRABAJO FUTURO</b>	<b>56</b>
<b>13</b>	<b>ANEXO 1: MANUAL DE INSTALACIÓN</b>	<b>57</b>
13.1	INSTALACIÓN DE WONDERLAND . . . . .	57
13.1.1	ACCESO A WONDERLAND DESDE OTROS EQUIPOS (OPCIONAL) . . . . .	64
13.2	INSTALACIÓN DE LA APLICACIÓN DEL PROFESOR . . . . .	65
13.3	CARGAR MODELOS DE GOOGLE SKETCHUP EN WON- DERLAND . . . . .	65
	<b>BIBLIOGRAFÍA</b>	<b>69</b>

# Índice de Tablas

3.1	Resumen de plataformas relacionadas con mundos virtuales . .	9
10.1	Actividades y duración de las mismas . . . . .	51
10.2	Tabla de salarios . . . . .	52
10.3	Tabla de recursos necesarios . . . . .	53
10.4	Presupuesto total . . . . .	53

# Índice de Figuras

3.1	Ejemplo de conexión . . . . .	10
3.2	Arquitectura de Wonderland . . . . .	11
3.3	Captura del cliente . . . . .	12
3.4	Arquitectura de la parte servidor . . . . .	13
4.1	Tejado del edificio . . . . .	16
4.2	Edificio . . . . .	17
4.3	Lateral del edificio . . . . .	17
4.4	Parte trasera del edificio . . . . .	17
4.5	Objeto creado a mano . . . . .	18
4.6	Objeto de Google Warehouse . . . . .	18
4.7	Aula . . . . .	19
5.1	Página principal de la aplicación para el profesor . . . . .	24
5.2	Ejemplo de archivo XML de Lengua . . . . .	25
5.3	Ejemplo de archivo XML de Matemáticas . . . . .	26
5.4	Ejemplo de archivo XML de Ciencias . . . . .	26
5.5	Ejemplo de registro . . . . .	27
6.1	Página principal de la aplicación addcontents . . . . .	28
6.2	Listado de alumnos . . . . .	29
6.3	Ejemplo de cómo añadir un alumno . . . . .	29
6.4	Alumno añadido . . . . .	30
6.5	Ejemplo de inserción de información . . . . .	30
6.6	Lista de contenidos . . . . .	31
6.7	Ejemplo de cómo editar información . . . . .	31
6.8	Lista de contenidos actualizada . . . . .	31
6.9	Ejemplo de añadir contenido . . . . .	32
6.10	Ejemplo de añadir contenidos . . . . .	32
6.11	Ejemplo de añadir contenido . . . . .	33
7.1	Aplicación de Matemáticas . . . . .	36
7.2	Aplicación de Lengua . . . . .	37



7.3	Aplicación de Ciencias . . . . .	38
8.1	Pantalla principal . . . . .	39
8.2	Ventana de login . . . . .	40
8.3	Pantalla del cliente . . . . .	40
8.4	Rincones dentro del aula . . . . .	41
8.5	Login de la aplicación . . . . .	41
8.6	Juego de Matematicas . . . . .	42
8.7	Juego de Lengua . . . . .	43
8.8	Juego de Ciencias . . . . .	44
13.1	URL de Wonderland . . . . .	59
13.2	Acceso a Wonderland desde un navegador . . . . .	59
13.3	Pantalla principal . . . . .	60
13.4	Opción "Manage Modules" del menú . . . . .	61
13.5	Eliminar un módulo . . . . .	61
13.6	Instalación de un módulo . . . . .	62
13.7	Selección de un módulo . . . . .	62
13.8	Proceder a instalar . . . . .	62
13.9	Confirmación de la instalación . . . . .	62
13.10	Pantalla de login . . . . .	63
13.11	Nueva casilla del menú . . . . .	63
13.12	Usuarios por defecto en Wonderland . . . . .	63
13.13	Pantalla de edición de la información del usuario . . . . .	64
13.14	Ventana para cargar el módulo . . . . .	66
13.15	Ventana para seleccionar el modelo . . . . .	67
13.16	Resultado final al cargar un modelo en Wonderland . . . . .	67

## Capítulo 1

# INTRODUCCIÓN

Este proyecto intentará afrontar el problema del colegio público Seseña y Benavente de Getafe ha tenido dificultades a la hora de mejorar sus aulas de informática, tanto el software como el hardware debido a su bajo presupuesto. Este proyecto va a servir para realizar una aplicación educativa la cual se adapte a las necesidades del centro.

Éste colegio tiene un aula de informática de 12 puestos para atender a casi 600 alumnos con una dotación informática del año 2002. Los ordenadores que utilizan son muy antiguos y es casi imposible adaptarlos a las nuevas tecnologías. En cambio, los padres de los alumnos compran máquinas muy potentes a sus hijos con las que podrían trabajar por lo que es interesante aprovechar este dato para la elección de una solución.

Los alumnos de primaria son nativos digitales debido a que desde edades más tempranas han estado rodeados de tecnología, ya sea un ordenador o una consola. Esto quiere decir que un alumno sabrá adaptarse fácilmente a una nueva tecnología.

La aplicación que se desarrollará para el tendrá que ser entretenida y educativa a la vez y además no requerirá coste de licencias. Esto implica que la aplicación tenga que ser de un modo u otro, un juego atractivo a la vista, con diferentes contenidos que puedan variar, para que el alumno se centre en la actividad que esté realizando. El juego no tendrá demasiados requisitos gráficos y estará desarrollado en software libre.

Desde el punto de vista del profesor, de nada sirve que el alumno utilice una aplicación si el profesor no puede influir en ella y comprobar la destreza que presenta el alumno. Por ello, el profesor deberá tener otra aplicación donde pueda manipular información utilizando archivos que utiliza comunmente

como pueden ser documentos, imágenes, audios o videos. También hay que añadir que los profesores a los que va dirigida la aplicación (primeros cursos de primaria) no suelen tener muchos conocimientos avanzados de nuevas tecnologías por lo que implica que la aplicación tenga que ser intuitiva y que el profesor pueda adaptarse fácilmente.

## **AMBIENTES DE APRENDIZAJE**

Para un niño de primaria, el ambiente o entorno donde realice su actividad influirá en el aprendizaje de la misma [1]. En el aula donde los alumnos realizan su actividad está dividida en los llamados rincones. Cada uno de los rincones están asociados a una actividad (lengua, matemáticas, conocimiento del medio o cualquier otra). De esta manera, el alumno asocia una localización física con un tipo de actividad. Esta idea también interesa a los padres ya que quieren saber en que ambiente se desarrolla el aprendizaje su hijo. Además de estos ambientes, en el aula pueden recrearse ambientes fantásticos relacionados con eventos del año como por ejemplo Navidad, donde realizando ciertas actividades se consiguen ciertas recompensas. Este otro tipo de ambiente que motiva al alumno no se tratará en este proyecto pero es necesario diseñar la aplicación de modo que pueda incluirse en un futuro.

Centrándose en la aplicación que se va a realizar, se recreará un entorno real lo más fiel posible al entorno escolar quedando abierta la posibilidad de crear otro tipo de entornos. Este entorno contendrá juegos sencillos simulando los rincones que podrán ser configurables y ampliados por el profesor y que además mantendrán de la actividad realizada.

### **DESDE EL PUNTO DE VISTA TÉCNICO:**

Existen muchos entornos virtuales y juegos para el aprendizaje, que se explicarán más adelante dando una visión general y se explorarán sobretodo Wonderland, orientado a colaboración y Unity3D, orientado a juegos.

Se buscará una solución que consiga modificar archivos por parte del profesor creando archivos de configuración basados en xml y que pueda tener la posibilidad de guardar el registro de los alumnos. Estos requisitos se usarán para la búsqueda de una solución pero en el proyecto se verificarán:

- La capacidad de leer/escribir ficheros de configuración.
- Carga computacional.

- Facilidad de instalación / configuración.
- Facilidad de integración con otras herramientas.

Además de explorar el interés de comunicar varias herramientas entre sí.

## Capítulo 2

# OBJETIVOS

Los objetivos de este proyecto desde el punto de vista del usuario final son:

1. Crear un mundo virtual 3D, en este caso, un colegio que sirva como portal para que los usuarios accedan a un aula donde se encontrarán las distintas áreas temáticas. Este mundo será una recreación del mundo real.
2. Reducir la dificultad de creación de contenidos por lo que el profesor, manejando ficheros sencillos como archivos de texto, imágenes, vídeos o audio, pueda cambiar el contenido de las actividades presentadas en el mundo virtual y que se adapten a las necesidades del profesor.
3. Crear juegos en Unity para hacer más vistosos y agradables al alumno los diferentes contenidos que existan en el mundo.
4. Tener un registro de las actividades que realice el alumno dentro del mundo, tanto un histórico de actividades accedidas como una evaluación de resultados (aciertos, fallos en cada una de ellas).

Desde el punto de vista técnico se conseguirán los siguientes objetivos:

1. Creación de modelos 3D a partir de fotografías e información geométrica para crear réplicas del mundo real y enriquecerlas con otros objetos virtuales.
2. Desarrollo 2D de juegos interactivos que sean sencillos de utilizar, atractivos visualmente y configurables, donde se pueda seguir la actividad del alumno y así evaluarle.

3. Hacer pruebas de integración entre diferentes motores gráficos para incorporar los juegos dentro del escenario.
4. Diseño de ficheros XML de configuración y que puedan ser compartidos desde varias aplicaciones. Para ello habría que definir los mecanismos de lectura y escritura de dichos ficheros desde los diferentes motores gráficos que se utilicen para los escenarios y los juegos.

## Capítulo 3

# ESTADO DEL ARTE

### 3.1 CONCEPTOS GENERALES

En este apartado se explicarán brevemente los distintos conceptos que aparecerán a lo largo del proyecto.

- **Mundo virtual:** Un mundo virtual es la simulación de un mundo o un entorno donde diversas personas a través de sus respectivos avatares pueden interaccionar con los diversos elementos de su entorno y también pueden comunicarse entre ellos.
- **Avatar:** Es la representación virtual de una persona en un mundo virtual.
- **Non Player Character (NPC):** Un NPC es un avatar existente en un mundo virtual, no controlado por una persona, que forma parte del programa informático que controla el mundo.
- **Motor gráfico:** Un motor gráfico es la parte de un programa que se encarga de controlar, gestionar los gráficos 3D en tiempo real.
- **Objeto:** Un objeto es un elemento situado dentro de un mundo virtual. Puede ser una aplicación, una imagen, un video, un audio o cualquier otro elemento.
- **Propiedades del objeto:** Son características que tiene un objeto o se le pueden añadir, como por ejemplo cambiar el color del objeto, la posición, permisos, visibilidad...

Los siguientes conceptos sólo son aplicables a Wonderland:

- **Módulo:** Un módulo es equivalente a un "Plugin" o una "Aplicación" en otras plataformas y sirve para extender la funcionalidad de dicha(s)

plataforma(s). Un módulo puede incluir código, imágenes e incluso mundos virtuales, además de otros recursos.

- **Objeto:** En Wonderland un objeto también es la representación de un módulo en el mundo. Por ejemplo, si hay un módulo cuya funcionalidad es representar una imagen en el mundo, un objeto sería cada uno de los elementos dispuestos en el mundo que contiene cada una de las imágenes.
- **Celda:** Se puede definir celda como la parte gráfica de un objeto (si la tiene) con la que un usuario puede interaccionar.

### 3.2 MUNDOS VIRTUALES

En la actualidad existen diversos simuladores de mundos virtuales por lo cual, se destacarán las características de los simuladores más importantes.

En Junio de 2003 apareció una de las plataformas más conocidas relacionada con mundos virtuales conocida por *SecondLife* (*SL*) [2], que posee un código abierto y libre programable en C++. *SL* utiliza una arquitectura cliente-servidor, donde el cliente es una aplicación que puede ser descargada y utilizada por cualquier usuario y mediante un avatar se comunicarán con los demás usuarios; mientras que el servidor es único, está centralizado y recibe las conexiones de todos los usuarios.

Los programadores pueden realizar sus propias aplicaciones y añadirlas a *SL*. Estas aplicaciones pueden distribuirse libremente o pueden venderse tanto por dinero real como por dinero virtual (moneda propia de *SL*).

Una limitación de *SL* es la realización de un mundo virtual propio ya que no es gratuito. *SL* vende "terrenos virtuales" para construir edificios propios y dependiendo del tamaño del terreno, su precio puede oscilar entre 100\$ y 1000\$ mensuales, aunque se puede pagar mediante dinero virtual. Esta es una forma de sacar beneficio económico de la plataforma. El otro beneficio es simplemente la colocación de banners publicitarios.

En 2007 aparece otra plataforma similar llamada *OpenSim* [16]. *OpenSim* se puede combinar con *SL* ya que utiliza la misma infraestructura de comunicación e incluye una mejora que es poder realizar simulaciones desde nuestro propio ordenador sin necesidad de incluirlo en la plataforma *SL* para poder probarlo. Esto da un extra de fiabilidad al usuario ya que no depende del servidor centralizado de *SL*. La ventaja de *OpenSim* es que se puede crear un mundo virtual propio y añadirlo a *SL* como una aplicación más. Gracias a



esto, un usuario puede evitar pagar los altos costes de "alquiler" de un terreno en SL. Esta ventaja que da *OpenSim*, repercutió en los precios de SL siendo estos inferiores a \$100 mensuales los terrenos más grandes. Esta plataforma es de código libre y programable en C++. Cabe destacar que *OpenSim* tiene un gran apoyo por parte de IBM.

En 2006 aparece *Wonderland*[3], que fue creado por Sun Microsystems y estaba orientado a los negocios. En Abril de 2009, Sun Microsystems fue vendido a Oracle y a principios del 2010, dejó de financiar económicamente la plataforma. Muchos usuarios que la utilizaban decidieron continuar por su cuenta el proyecto haciéndose independientes de Oracle y cambiando el nombre de la plataforma por *Open Wonderland*. Poco después las empresas que utilizaban esta plataforma ofrecieron apoyo económico hacia esta iniciativa iniciada por los usuarios por lo que *Wonderland* deja de tener un futuro incierto.

Wonderland está programado en Java, y su código es completamente libre (GPLv2, BSD). Wonderland utiliza la arquitectura de cliente-servidor, donde ambos están incluidos en el mismo paquete. En esta plataforma, cada usuario puede crear su propio mundo virtual, ejecutarlo desde su propio ordenador y a través de una URL, puede compartir su mundo, por lo que no es necesario descargar ningún programa adicional para conectarse a un mundo. Además posee un potente módulo de voz donde los usuarios pueden comunicarse en tiempo real. Wonderland tiene una desventaja respecto a las otras dos plataformas, ya que comparada con ellas, es muy lenta y necesita una tarjeta gráfica más potente y una buena conexión a internet si se desea utilizar el módulo de voz. De igual manera que en *SL*, un usuario que realice una aplicación (módulo), puede distribuirla libremente o venderla.

Una de las principales razones por la que *Wonderland* es lenta respecto a las otras dos plataformas anteriores es por el motor gráfico que utiliza. *SL* y *OpenSim* utilizan OGRE (Object-Oriented Graphics Rendering Engine), un motor gratuito programado en C++ y muy potente preparado para mostrar gráficos en tiempo real; en cambio, *Wonderland* utiliza JME (JMonkeyEngine) un motor programado en java, el cual no está preparado para mostrar gráficos a tiempo real, sino con una cierta tasa por lo que influye notablemente a la hora de moverse por un mundo virtual. Otros simuladores conocidos como Croquet, Multiverse o RealXtend están resumidos en la tabla a continuación.

Además de estas plataformas pensadas para mundos virtuales existen otras

Plataforma	Página Web	Desarrollado por	Arquitectura	Lenguaje de desarrollo	Motor gráfico
SecondLife	<a href="http://secondlife.com/">http://secondlife.com/</a>	Linden Lab	cliente-Servidor	C#	Ogre
OpenSim	<a href="http://opensimulator.org">http://opensimulator.org</a>	IBM(?)	cliente-Servidor	C#	Ogre
Wonderland	<a href="http://www.openwonderland.org/">http://www.openwonderland.org/</a>	Java.net	Cliente-Servidor	Java	JME
Multiverse	<a href="http://www.multiverse.net">http://www.multiverse.net</a>	Multiverse	Cliente-Servidor	Python	Ogre
Croquet	<a href="http://www.opencroquet.org">http://www.opencroquet.org</a>	The Croquet Consortium	Peer-2-peer	Squeak	OpenGL

**Tabla 3.1** – Resumen de plataformas relacionadas con mundos virtuales

aplicaciones como Unity3D [18], que es un motor de videojuegos con posibilidad de crear juegos multijugador, donde se podría emular un mundo virtual. De esta manera se podrían crear mundos virtuales propios a partir de cero con las características que uno desee, a cambio de mayor complejidad en su desarrollo.

Los requisitos principales que debe cumplir cualquier mundo virtual según [15] son los siguientes:

1. Cualquier persona en un mundo virtual tiene que poder ver las mismas cosas que cualquier otra persona conectada en ese mismo mundo. *Por ejemplo, si dos personas se encuentran en una sala, si una ve un libro en un sitio, la otra debe verlo en el mismo sitio.*
2. Si algo cambia dentro del mundo, todos los usuarios deberán tener actualizada su visión del mundo. *Si una de las dos personas se lleva el libro, la otra deberá ver que el libro ya no está ahí.*
3. Los usuarios se deben poder conectar cuando lo deseen y tener actualizado el mundo en el momento de la conexión. *Si un usuario se desconecta por la razón que sea y se vuelve a conectar más adelante, si la persona que se llevo el libro lo deja en otro sitio mientras el otro estaba desconectado, este usuario deberá ver el libro en su nueva localización.*
4. Los usuarios no necesitan saber la localización física de un usuario en el mundo real. *Los usuarios no necesitan saber desde donde se conectan los demás, ni necesitan comunicarse directamente desde sus ordenadores.*

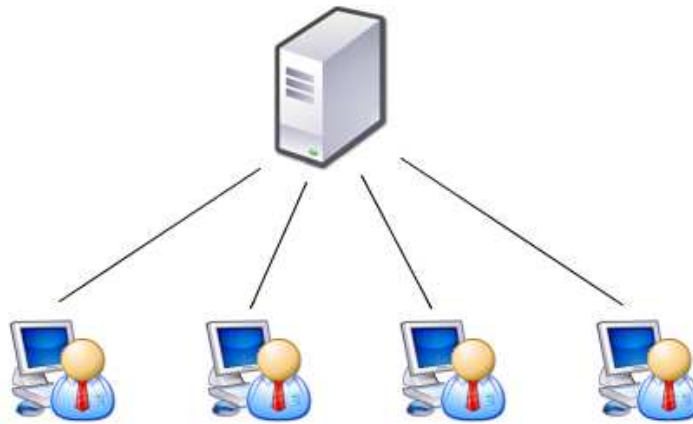
### 3.3 WONDERLAND

En el apartado anterior se explicó brevemente algunas de las características de la plataforma *Wonderland*. Puesto que va a ser la plataforma elegida para la realización de este proyecto, a continuación, se explicará con más detalle su arquitectura, sus características y su funcionamiento.

### 3.3.1 QUÉ ES WONDERLAND

*Wonderland* es una plataforma de código libre, programada en Java para la simulación de mundos virtuales, su motor gráfico es JME también programado en Java.

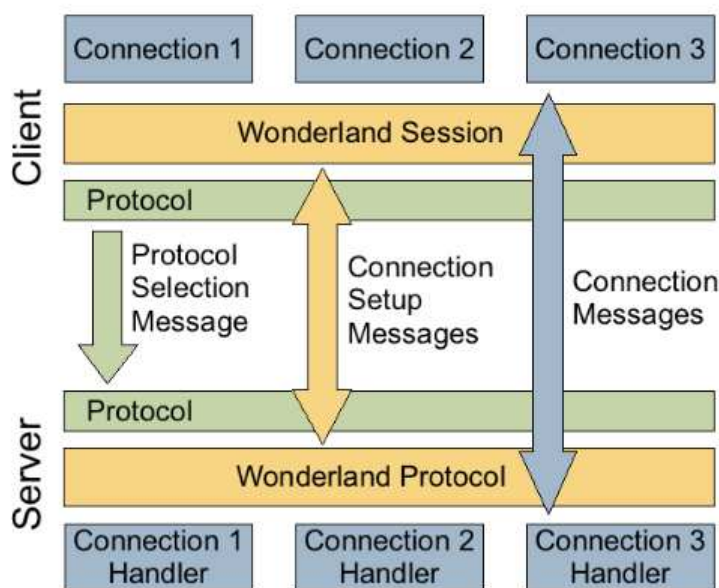
Para un usuario final, *Wonderland* será como un programa en el que se pueden conectar varias personas en un mundo virtual, pudiéndose comunicar entre ellos como si fuese una red social, utilizar aplicaciones o manipular, añadir o editar archivos de audio, video o texto. Todos los usuarios conectados a un mismo mundo dependen de un nodo central, el cual gestiona y procesa toda la información utilizada por los usuarios.



**Figura 3.1** – Ejemplo de conexión

### 3.3.2 ARQUITECTURA

La arquitectura de *Wonderland* es de cliente-servidor y esta dividida en tres capas de comunicación: cliente, "common" y servidor.



**Figura 3.2** – Arquitectura de Wonderland

- **Cliente:** El cliente es la parte gráfica del mundo virtual (interfaz de usuario). Para arrancar el cliente se necesita una URL proporcionada por el servidor *Wonderland*. Esta URL suele ser la IP de la máquina física en la que está ejecutándose el servidor. Una vez que el usuario entra en el mundo virtual se pueden ver los avatares de los demás usuarios, los módulos insertados en él y diversas opciones en la que se le permite añadir nuevos módulos, compartir imágenes o aplicaciones, etc. En la figura 3.3 se puede ver una captura del cliente.

Algunos módulos o aplicaciones tienen ciertas limitaciones a la hora de manipularlos ya que se pueden establecer permisos o propiedades a cada uno de ellos. El creador del mundo virtual puede añadir propiedades independientemente para los diversos elementos del mundo. Estas propiedades pueden ser de manipulación (rotación, escala o translación de los elementos), también puede decidir que otros usuarios tienen permiso para añadir o no elementos al mundo, si existe colisión entre los distintos elementos del mundo, etc.

Esto quiere decir que si por ejemplo, el creador del mundo coloca una imagen que no quiere que sea manipulada, debe establecer un permiso para que otros usuarios no puedan cambiar sus propiedades (posición,

orientación y tamaño). Si no es así, cualquier usuario puede manipularla e incluso eliminarla si lo desea.



Figura 3.3 – Captura del cliente

La comunicación entre usuarios puede ser de dos tipos: por texto o por voz. La comunicación por texto suele ser a través de un chat donde cada uno de los usuarios puede introducir información que será recibida por los demás usuarios o restringir el acceso a un único usuario o grupo. En cambio, para la comunicación por voz se necesita un micrófono y una conexión de más capacidad para enviar y escuchar los mensajes con claridad.

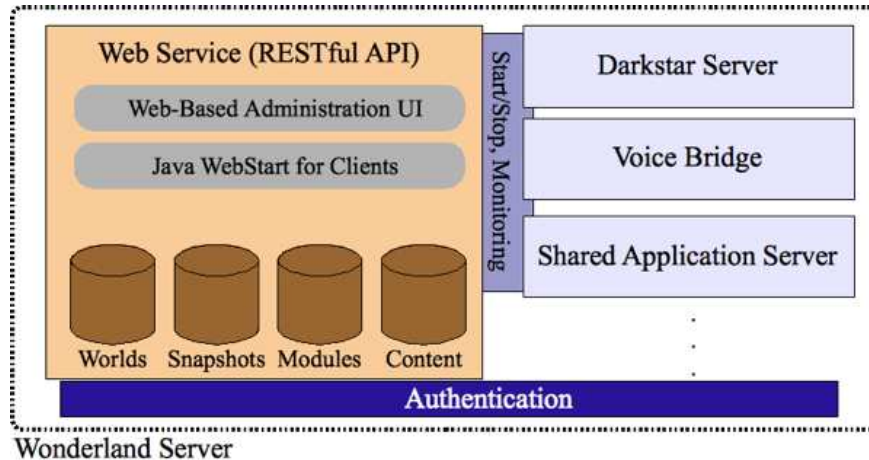
- **Common:** Es la parte común y sirve para establecer conexiones entre cliente y servidor y enviar mensajes en cualquiera de los dos sentidos para notificarse entre sí los cambios que se produzcan.

Si un usuario manipula el tamaño de una imagen, esta manipulación debe comunicarse a los demás usuarios y además, se deben guardar los cambios. Para ello, se utilizan los mensajes. El mensaje lleva la información del cambio hacia el servidor y el servidor se encarga de guardar el nuevo estado de la imagen y se envía este mensaje a todos los usuarios para que puedan ver el cambio en el mundo.

Por defecto, no es necesario configurar ningún protocolo para la comunicación puesto que *Wonderland* utiliza uno propio para hacerlo. Si

se necesitase añadir algún protocolo adicional se realizaría en la parte común.

- **Servidor:** El servidor es la parte donde se almacena toda la información del mundo virtual. En él se encuentran todos los módulos instalados, los escenarios de los mundos creados, información del estado del servidor, aplicaciones externas que se han incluido, puntos de posición existentes e información de los contenidos.



**Figura 3.4** – Arquitectura de la parte servidor

### 3.4 UNITY3D

Unity3D [18] es una plataforma gratuita para la creación de videojuegos bajo Windows y Mac. Inicialmente era una plataforma de pago pero a partir de Enero de 2010 comenzó a distribuirse también una versión gratuita con gran parte de la funcionalidad original. Los lenguajes de programación que pueden utilizarse para programar en esta plataforma son: Javascript, C++, Android y Boo (derivado de Python). Pueden realizarse diferentes tipos de juegos según la perspectiva del usuario (en primera persona o tercera persona) y según la interfaz gráfica (2D o 3D).

Esta plataforma funciona sobre Windows y Mac OS, pudiéndose exportar los juegos como una aplicación de Windows, Mac, Wii, iPhone, iPad, Xbox 360, Playstation 3 o una aplicación web; aunque alguna de las plataformas específicas (no PC) requieren licencia de pago.

Unity3D puede utilizar varios motores gráficos que son Direct3D (Windows), OpenGL (Mac y Windows) OpenGL ES (iPhone) y otros propietarios como pueden ser los de la Wii.

Esta plataforma se caracteriza por los llamados "Assets". Los Assets son elementos ya predefinidos como puede ser un objeto 3D que se puede importar dentro de un juego. Con estos Assets se pueden diseñar entornos de cualquier tipo. Unity3d también tiene soporte multijugador, por lo que si se crea un escenario en el que se puedan conectar varios usuarios desde diferentes ordenadores, se puede simular un mundo virtual en el que interactuen todos ellos.

### 3.5 XML

XML (eXtensible Markup Language) [4] es un lenguaje de etiquetas desarrollado por el World Wide Web Consortium (W3C) ?? que se utiliza como estándar para el intercambio de información entre diferentes plataformas.

Al ser un lenguaje extensible se pueden añadir etiquetas siempre que sea necesario.

Las partes que tiene un documento XML son:

Prólogo: Describe la versión del XML, tipo de documento, etc. (Opcional)

Cuerpo: Es el elemento raíz del documento. (Obligatorio)

Elementos: Los elementos pueden tener contenido o ser elementos vacíos. (Opcional)

Atributos: Sirven para incorporar características a los elementos y deben ir entre comillas. (Opcional)

Entidades predefinidas: Entidades para representar caracteres para que no sean interpretados como marcado en el proceso XML.

Secciones CDATA: Para especificar datos que no sean procesados por el proceso XML.

Comentarios: Para incluir líneas de modo informativo e ignorador por el procesador.

## Capítulo 4

# DISEÑO DE ESCENARIOS

En este apartado se explicará el proceso de creación de los modelos de escenario (edificio y aula) que conforman el mundo virtual diseñado para este proyecto. De esta manera, cualquier usuario que se conecte a éste mundo desde su PC usando un cliente Wonderland, podrá desplazarse fácilmente y podrá ver la réplica virtual del colegio sin moverse de su casa.

### 4.1 TOMA DE DATOS

Uno de los requisitos de la aplicación a desarrollar era que el mundo virtual fuese una réplica lo más fiel posible del mundo real. Para ellos se solicitó una autorización al colegio para la toma de fotos y medidas in situ. Un profesor y un encargado del colegio facilitaron unos planos con las medidas y un horario de acceso para tomar fotos tanto del exterior del edificio como del interior del aula.

Una vez tomados los datos necesarios, se procede a crear el edificio y el aula.

Para la creación del edificio y del aula se decidirá utilizar la aplicación Google Sketchup.

Google Sketchup [7] es una herramienta gratuita con la que se pueden crear modelos 3D de forma fácil e intuitiva y sin necesidad de tener conocimientos previos de herramientas de diseño gráfico.

Los modelos creados en Google Sketchup tienen la ventaja de que su formato está soportado por otras muchas plataformas (Unity3D, Wonderland, SecondLife, Opensim...) lo que facilita su reutilización si es necesario cambiar de motor gráfico..

La principal razón para replicar exactamente el mundo real en el entorno vir-

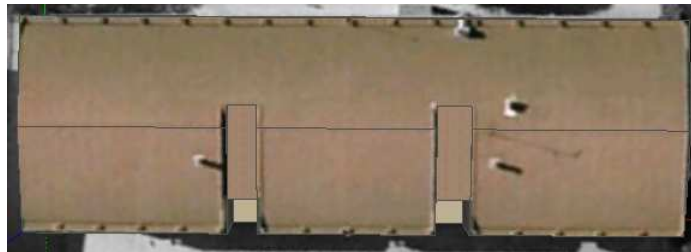


tual era facilitar la navegación del alumno. Esto es muy importante puesto que se trata de alumnos muy pequeños (Primero y segundo curso de primaria). Al tratarse de un entorno conocido, la primera vez que el alumno entra en la aplicación visualizaría el edificio, sabría identificar los elementos que lo componen y se dirigiría hacia él sin necesidad de instrucciones previas. Una vez en el edificio, aparecería dentro de un aula en el que estarían incluidas las aplicaciones. Cada una de estas aplicaciones es una zona específica que simulan los rincones de trabajo habituales en el entorno real y desde allí podría acceder a las aplicaciones correspondientes (lengua, matemáticas o ciencias).

## 4.2 DISEÑO DEL EDIFICIO

Para realizar el diseño del edificio se utilizaron dos fuentes de información para poder asemejar el diseño lo más posible a la realidad:

- a) Los planos de edificio proporcionados por el encargado del colegio. El principal problema de los planos es que únicamente incluyen el ancho y el alto del edificio por lo que la altura debe ser estimada.
- b) Google Earth, que nos proporciona igual que los planos, el alto y el ancho de los edificios, pero adicionalmente proporciona una imagen del tejado que no podría obtenerse con las fotos.



**Figura 4.1** – Tejado del edificio

Google Earth [6] es una aplicación de Google libre que permite visualizar cualquier zona del mundo y ver imágenes de satélite, mapas, relieves y edificios 3D. Una de las ventajas de utilizar esta aplicación es la posibilidad de importar el plano de Google Earth dentro de Google Sketchup con sus medidas reales. Igualmente se desconoce la altura del edificio pues como en el punto a), hay que estimarla.

Una vez realizado el modelo se utilizan las fotos de la fachada con algunas modificaciones para añadir al texturas que dotarán al modelo con mayor realismo.

De este modo se consigue el exterior del edificio.



**Figura 4.2** – Edificio



**Figura 4.3** – Lateral del edificio



**Figura 4.4** – Parte trasera del edificio

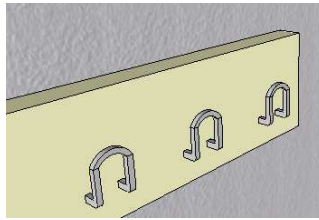
### 4.3 DISEÑO DEL AULA

El diseño del aula difiere un poco del edificio, por dos motivos:

- a) En este caso las medidas del aula se desconocen y no son tan relevantes ya que prima la facilidad de movimiento del alumno en el aula y la disposición de los elementos igual que en la realidad (ubicación de los rincones de trabajo, pizarra, estantería, etc).

- b) En este caso nos conviene usar las fotos como textura porque dentro del aula existen muchos objetos y muchos de ellos no se pueden mover. Además las fotografías contienen elementos como mesas, armarios y sillas que, si se "pegan" en las paredes del aula, quedarían poco realistas. En cambio, el exterior del edificio no tiene "ningún" elemento que tape la textura principal, aunque puede existir algún árbol o alguna valla, son elementos que ocupan muy poco espacio en comparación con el tamaño del edificio.

Por estar 2 razones se decidirá realizar el modelo del aula totalmente a mano, creando desde cero la mayor parte de los elementos existentes en el aula (muebles, mesas, percheros) y utilizando elementos prefabricados de Google Warehouse (sillas, pizarra, radiadores). Para las paredes y el suelo sí se utilizaron las texturas de las fotografías, ya que es sencillo extraer una baldosa o un trozo de pared.



**Figura 4.5** – Objeto creado a mano

Google Warehouse [8] es una galería donde los usuarios suben sus modelos creados en Google Sketchup de forma que otros usuarios puedan descargárselos libremente.



**Figura 4.6** – Objeto de Google Warehouse

El resultado final del aula sería el siguiente:



**Figura 4.7** – Aula

## Capítulo 5

# APLICACIÓN PARA EL PROFESOR

La aplicación para el profesor es una aplicación donde el profesor podrá insertar, editar, modificar o borrar información relacionada con diferentes temáticas (lengua, matemáticas o ciencias).

El principal requisito de la herramienta del profesor es que sea una aplicación intuitiva y fácil con la que se puedan manipular ficheros sencillos y conocidos como son imágenes, archivos de texto, videos o audios. Esto es así porque la mayoría de los profesores que la usarán son personas con pocos conocimientos informáticos.

### 5.1 ANALISIS DE OPCIONES

A continuación se describen las posibles soluciones analizadas, sus ventajas e inconvenientes y la opción elegida tras las pruebas de implementación.

1. Crear una **aplicación en la parte cliente dentro del mundo virtual utilizando librerías externas** para almacenar información utilizando Google Docs. Las ventajas de esta idea son:
  - La posibilidad de manipular la información dentro y fuera del mundo virtual:
    - Dentro: Creando una aplicación que se comunique con Google Docs.
    - Fuera: Entrando en la cuenta de Google Docs y manipular desde ahí la información

Tras implementar esta solución, los problemas encontrados fueron los siguientes:

- Creación de una cuenta de correo de Google. Para manipular la información se necesita crear una cuenta de correo externa

a la aplicación, que aunque no tenga mucha dificultad, requiere conocimientos extra por parte del profesor.

- Pérdida de flexibilidad. Los ficheros que se pueden manipular con Google Docs están limitados a documentos de texto, hojas de cálculo, presentaciones, etc. El manejo de videos o audios sería más complejo, sobre todo, a la hora de extraer la información de los documentos y es uno de los requisitos de partida.
- Seguridad de los documentos. Google Docs da la posibilidad de que los documentos sean públicos, privados o restringidos a cierto número de usuarios. Si los documentos son privados, el profesor podrá manipular los documentos sin ningún problema pero las aplicaciones de los alumnos que acceden a dichos documentos no podrán extraer la información. Si los documentos son restringidos a ciertos usuarios, los alumnos deberían crearse una cuenta de Google o ser creada por el profesor, por lo que puede ser bastante complejo y dar problemas de seguridad al tratarse de alumnos pequeños. Y por último, si los documentos son públicos, cualquiera podría acceder a ellos y manipular la información.
- Visibilidad. Los demás usuarios en el mundo podrían ver toda la actividad que realice el profesor con su aplicación.

2. Crear una **aplicación en la parte servidor con librerías de Google Docs**. Tras implementar esta opción se observa que las condiciones mejoran considerablemente con la anterior debido a:

- Se tienen las mismas ventajas que en primer caso.
- Aumenta la velocidad de acceso a los distintos apartados de la aplicación ya que la parte del servidor no necesita cargar la cantidad de gráficos que tiene la parte cliente.
- Mejora la visibilidad y la seguridad. No se tiene el problema de visualización de la actividad del profesor ya que en la parte servidor sólo se puede acceder a través de un usuario y una contraseña; y mejora la seguridad debido a que no hay que añadir permisos para evitar su manipulación aunque se sigue teniendo el problema del acceso al contenido.

Las desventajas son las mismas que el caso anterior exceptuando la visibilidad, que en este caso son la creación de una cuenta de correo Google, la pérdida de flexibilidad, la seguridad de los documentos y debido a la estructura del servidor, se está limitado a realizar un Servlet.

3. Crear una **aplicación en la parte cliente sin librerías externas**. Para probar este caso se hizo uso de las librerías propias de Wonderland creando para ello un módulo. Las ventajas encontradas fueron las siguientes:

- No se necesitan librerías externas a Wonderland, por lo que se simplificaría su programación.
- Los documentos se almacenarían dentro del directorio de Wonderland, por lo que la lectura y escritura de dichos documentos sería más sencilla.
- A diferencia de la opción anterior, sería más sencillo trabajar con ficheros multimedia debido a que la información no tiene que tener una estructura determinada.

Las desventajas encontradas fueron las siguientes:

- Seguridad. Debido a que la aplicación está en la parte cliente, cualquier usuario podría acceder a la aplicación, por lo tanto, habría que añadir seguridad para que el objeto solo sea manipulable por el profesor (tamaño, forma o situación); y seguridad para que solo pueda acceder él, como un usuario y una contraseña.
- Visibilidad. Aunque la aplicación sea segura y solo el profesor sea responsable de su uso, los demás usuarios en el mundo podrían ver toda la actividad que realice el profesor con su aplicación, igual que ocurría en el caso 1.

4. Crear una **aplicación en la parte servidor dentro del mundo virtual**. En este caso se probó crear un Servlet incluido como una opción más en el menú del servidor. Las ventajas de hacerlo de esta manera son las siguientes:

- Se trabaja únicamente con la parte servidor de Wonderland, por lo que la velocidad de acceso a las diferentes opciones de la aplicación mejora considerablemente con respecto la opción de realizarlo en la parte cliente debido a que no necesita requisitos gráficos.
- Incluye todas las demás ventajas de la tercera opción.
- Seguridad. Mejora la seguridad considerablemente debido a que solo puede acceder el profesor con un usuario y una contraseña sin necesidad de añadir permisos a la aplicación.

- Visibilidad. La aplicación solo puede ser visualizada por el profesor y quien tenga permiso a la misma.

Desventajas:

- Está limitado a la creación de servlets.

Finalmente y tras las pruebas realizadas se ha optado por no añadir una aplicación externa a Wonderland (con o sin librerías de Google Docs) debido a que el profesor tendría que aprender a manejar la aplicación y Wonderland independientemente. Estando la aplicación integrada en la plataforma Wonderland, simplificaría el uso de la aplicación y de Wonderland.

Un ejemplo sencillo sería el siguiente. El profesor quiere poner un mundo con unos edificios y quiere que los documentos tengan cierta información. En el primer caso, el profesor deberá utilizar la aplicación externa para incluir la información y después, acceder a Wonderland y añadir los contenidos y guardar el estado del mundo. Para ello necesita aprender como se maneja la aplicación y como se maneja Wonderland debido a que son 2 aplicaciones completamente diferentes. En el segundo caso, si la aplicación esta integrada en Wonderland, el profesor añade la información que desea y sin necesidad de abrir otra aplicación, manipular el mundo. La ventaja es que la aplicación integrada tiene la misma estructura que cualquier otra aplicación existente en Wonderland, por lo tanto, saber manejar la aplicación facilita el uso de las demás.

## 5.2 IMPLEMENTACIÓN

Tras las pruebas de implementación descritas en el apartado anterior se decidirá optar por la solución 4 "Crear una aplicación en la parte del servidor sin utilizar librerías externas".

Para el desarrollo de la aplicación se han utilizado las siguientes herramientas:

Subversion: Se ha utilizado Subversion (en adelante SVN) [17] para dos cosas:

- a) Acceder a las diferentes versiones del repositorio Wonderland con la que se trabajará. Es recomendable descargar la última versión disponible de esta web [14].
- b) Para la entrega y mantenimiento del código desarrollado para este proyecto.



ANT: Herramienta para compilar y ejecutar el código de Wonderland [5].

NetBeans: Es una plataforma en la cual se pueden desarrollar aplicaciones java y es con la que se trabajará en este proyecto [11].

Para hacerlo, se utilizarán Servlets, páginas JSP y programación en Java.

Un Servlet [9] es un programa que se ejecuta en un servidor web, que permite la interactividad del usuario y su uso más común es generar contenido para páginas webs de forma dinámica.

JSP (Java Server Pages) [10] es una tecnología que permite generar contenido para páginas web de forma dinámica. También sirve para generar Servlets.

### 5.3 DISEÑO DE JUEGOS Y ARCHIVOS DE CONFIGURACIÓN

Como ya se mencionó anteriormente, la aplicación tiene que ser lo más sencilla posible para que un profesor con poca experiencia informática. Desde la herramienta el profesor debe ser capaz de crear ejercicios relacionados con tres temas distintos (lengua, matemáticas y ciencias) que se ubican en los tres rincones existentes para tal fin en el aula física y en su réplica virtual.

Para hacer esto se diseñará para el alumno una aplicación con 3 juegos (uno por cada categoría y un fichero de configuración por cada uno de ellos). En la parte servidor el profesor deberá ser capaz de crear o modificar ficheros donde se almacenará la información. Eso quiere decir que tendremos tres archivos los cuales se llamarán lengua.xml, matematicas.xml y ciencias.xml. El archivo se crea la primera vez que se introduce un dato en cualquiera de las categorías y después, solo se actualiza el archivo.

#### Contenidos

Temática	Acciones	Registro de alumnos
Matemáticas	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Lengua	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Ciencias	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Lista de alumnos	<a href="#">Editar Lista</a>	

**Figura 5.1** – Página principal de la aplicación para el profesor

Estos archivos creados son los llamados "archivos de configuración" que son los que se utilizarán para intercambiar información entre las diferentes aplicaciones.

En la categoría relacionada con la temática de **Lengua**, el juego consiste en que a partir de un conjunto de letras crear tantas palabras como sea posible. Para ello el profesor deberá crear listas de palabras. Deberá introducir una palabra principal y todas las palabras que se puedan formar a partir sus letras. Por cada lista nueva que se introduzca será un nivel adicional en el juego para el alumno. Por ejemplo, si se introduce la palabra "casa" como palabra principal, se deberá introducir todas las posibles palabras que se puedan formar a partir de la principal. Algunas de ellas serán "asa", "as" o "saca".

En la figura 7.2 se puede ver la pantalla del juego y en la figura 5.2 el archivo de configuración correspondiente.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content-list>
  <content>
    <list>1</list>
    <item>casa</item>
    <items>as</items>
    <items>saca</items>
  </content>
</content-list>
```

**Figura 5.2** – Ejemplo de archivo XML de Lengua

En la categoría de **Matemáticas**, el juego consistirá en un problema de descomposición de números en unidades decenas, centenas, etc. En este caso el profesor simplemente deberá de introducir un enunciado y la respuesta del enunciado, que deberá ser un número. Cada problema que se introduzca, igual que en la categoría de lengua, será un nivel adicional. Un ejemplo de esta categoría sería poner como enunciado "50 + 5" y como resultado "55".

En la figura 7.1 se puede ver la pantalla del juego y en la figura 5.3 el archivo de configuración correspondiente.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content-list>
  <content>
    <list>1</list>
    <item>100 + 30 + 3</item>
    <item2>133</item2>
  </content>
  <content>
    <list>0</list>
    <item>1000 + 500 + 10 + 1</item>
    <item2>1511</item2>
  </content>
</content-list>

```

**Figura 5.3** – Ejemplo de archivo XML de Matemáticas

Por último, la categoría de **Ciencias**, desde el punto de vista técnico es muy similar a la de lengua ya que consiste en relacionar asociaciones correctas, con la diferencia de que en vez de introducir palabras, se introducen imágenes. En este caso el juego consistirá en introducir una imagen con puntos de interés que el alumno deberá conocer. Para resolver el problema, el alumno tendrá que arrastrar una foto de una zona ampliada sobre el nombre correspondiente. Para ello el profesor deberá introducir una imagen principal con unas zonas de interés e imágenes relacionadas con las zonas de interés de la imagen principal. Por ejemplo, la imagen principal podría ser un esqueleto humano con puntos rojos en ciertos huesos, que serían las zonas de interés; y las imágenes relacionadas serían los huesos donde están situados los puntos rojos. Del mismo modo se puede utilizar para identificar distintas zonas en cualquier imagen. Por ejemplo, regiones en un mapa, partes del cuerpo, de una flor, etc.

En la figura 8.8 se puede ver la pantalla del juego y en la figura 5.4 el archivo de configuración correspondiente.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content-list>
  <content>
    <list>3</list>
    <item>file:///C:/Users/Selene/.wonderland-server/0.5-preview4/run/content/system/content/imagenes/costillas.jpg</item>
    <items>file:///C:/Users/Selene/.wonderland-server/0.5-preview4/run/content/system/content/imagenes/omoplato.jpg</items>
    <items>file:///C:/Users/Selene/.wonderland-server/0.5-preview4/run/content/system/content/imagenes/vertebra.jpg</items>
    <items-name>omoplato</items-name>
    <items-name>vertebra</items-name>
  </content>
</content-list>

```

**Figura 5.4** – Ejemplo de archivo XML de Ciencias

## 5.4 EVALUACIÓN DEL JUEGO

Otro de los requisitos del programa es que el profesor pueda evaluar la actividad del alumno. Para ello se ha creado en cada aplicación un apartado llamado "Registro".

En este registro se almacena la información de cada uno de los usuarios que utilicen las aplicaciones para tener un control de quién ha realizado las actividades, los errores cometidos y la nota obtenida. Gracias a esto, el profesor obtiene información de las actividades propuestas y puede tomar las decisiones necesarias para mejorar el aprendizaje.

### Registros

Alumno	Nota	Fallos	Nivel alcanzado
selene	10	0	2

**Figura 5.5** – Ejemplo de registro

Al tratarse de un mundo virtual de libre acceso, cualquier persona puede conectarse para ver el mundo, y por lo tanto, utilizar las aplicaciones sin necesidad de ser alumno, el profesor obtendría un feedback de todas las actividades realizadas y sería complejo filtrar los resultados de sus alumnos. Para ello existe una casilla llamada "Lista de alumnos", por lo que solo recibirá un feedback de los alumnos listados.

Los datos que se deben incluir en esta lista son el nombre del alumno y una contraseña para cada uno. El uso de la contraseña servirá para que cada alumno acceda a las aplicaciones con sus datos y así evitar que los alumnos accedan a las cuentas de sus compañeros.

## Capítulo 6

# MANUAL DEL PROFESOR

Éste manual explicará como se tiene que utilizar la aplicación del profesor. Para ello se necesita haber realizado previamente la instalación de Wonderland e instalar la aplicación llamada "addcontents" que es la que se ha realizado para facilitar la edición de los ficheros XML de configuración por parte del profesor.

Instalado Wonderland y la aplicación, se podrá observar que en el menú de la izquierda aparece una opción llamada "Contenidos". En esta sección se puede observar que existen tres temáticas predefinidas llamadas Matemáticas, Lengua y Ciencias y una opción además de las temáticas llamada "Lista de alumnos".

### Contenidos

Temática	Acciones	Registro de alumnos
Matemáticas	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Lengua	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Ciencias	<a href="#">Añadir contenido</a> <a href="#">Ver contenido</a>	<a href="#">Registro</a>
Lista de alumnos	<a href="#">Editar Lista</a>	

**Figura 6.1** – Página principal de la aplicación addcontents

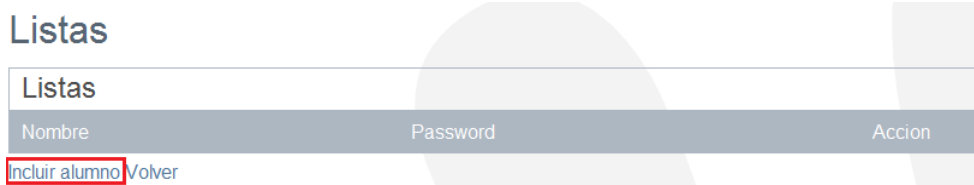
Cada temática tiene tres opciones llamadas "Añadir contenido", "Ver contenidos" y "Registro".

- **Añadir contenidos:** Esta opción permitirá añadir información a cada una de las temáticas. En cada temática se añadirá un tipo diferente de información que se explicará más adelante. Cada contenido añadido corresponderá a un nivel de juego distinto dentro del mismo

tema. Por ejemplo, si se añaden 3 contenidos en lengua, 1 en matemáticas y 5 en ciencias, las aplicaciones de lengua, matemáticas y ciencias tendrán 3, 1 y 5 niveles de juego respectivamente.

- **Ver contenidos:** Esta opción permitirá visualizar una lista de los contenidos introducidos de cada una de las temáticas. Desde esta sección se podrán modificar o añadir otros contenidos.
- **Registro:** Esta opción permite visualizar los resultados de los alumnos que han utilizado las aplicaciones. En ella se puede ver el número de aciertos, fallos y niveles alcanzados de cada una de las aplicaciones por cada alumno. Hay un registro independiente para cada temática.

La "Lista de alumnos" solo tiene una opción llamada "Editar Lista". Esta opción permite al profesor crear una lista de alumnos introduciendo un usuario y una contraseña para cada uno de ellos.

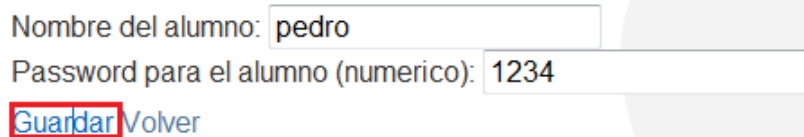


The screenshot shows a web interface titled "Listas". Below the title is a table with three columns: "Nombre", "Password", and "Accion". Below the table, there is a red box containing the text "Incluir alumno" and a blue link "Volver".

**Figura 6.2** – Listado de alumnos

El usuario y la contraseña se facilitarán al alumno para poder utilizar las aplicaciones. Este sistema permite que la evaluación se haga únicamente para los alumnos previamente registrados. De esta forma, se puede permitir el acceso a la aplicación a otras personas (padres, profesores, etc sin complicar la visión de los resultados por parte del profesor). También permitirá que entre los alumnos no accedan a la cuenta de sus compañeros.

## Incluir Lista



The screenshot shows a form titled "Incluir Lista". It has two input fields: "Nombre del alumno:" with the value "pedro" and "Password para el alumno (numerico):" with the value "1234". Below the fields, there is a red box containing the text "Guardar" and a blue link "Volver".

**Figura 6.3** – Ejemplo de cómo añadir un alumno

Una vez introducidos los datos del alumno, hay que pulsar la opción de "Guardar" para confirmar los datos. Añadido un alumno, aparecerá en la "Lista de alumnos" junto a una opción llamada "borrar" para eliminar al alumno en el caso de que se necesite.

## Listas

Listas		
Nombre	Password	Accion
pedro	1234	<a href="#">Borrar</a>

[Incluir alumno](#) [Volver](#)

**Figura 6.4** – Alumno añadido

Con esto se tiene un pequeño resumen del funcionamiento de la aplicación. A continuación se explicará más detalladamente el tipo de información que se puede introducir en cada temática y cómo hacerlo.

## 6.1 CONTENIDO DE MATEMÁTICAS

Para incluir contenidos a la aplicación de matemáticas, hay que pulsar en la opción "Añadir contenido" de esta temática. La página que aparece pedirá como información un enunciado y una solución. Se recomienda que el enunciado sea lo más breve posible para evitar problemas con la visualización del mismo en la aplicación para el alumno. La solución deberá ser numérica, porque en caso contrario, la aplicación que utilice el alumno va a ser incapaz de reconocer dicha información.

### Incluir Lista

Enunciado:	<input type="text" value="1510"/>
Solucion:	<input type="text" value="1000 + 500 + 10"/>

[Incluir lista](#) [Volver](#)

**Figura 6.5** – Ejemplo de inserción de información

Cuando se ha introducido la información deseada se pulsa en "Incluir lista". En el caso de haber dejado alguna casilla en blanco, se visualizará un mensaje

de error por pantalla informando que no se ha completado toda la información.

En el caso de que la información introducida sea errónea y haya que corregirla, hay que acceder a la opción de "Ver contenidos". En esta sección se pueden ver las listas incluidas hasta el momento pudiéndose modificar o eliminar.

Niveles		
ID	Contenido	Acciones
1	100 + 30 + 3	<a href="#">Editar</a> <a href="#">Borrar</a>
0	1000 + 500 + 10	<a href="#">Editar</a> <a href="#">Borrar</a>

[Volver](#)

**Figura 6.6** – Lista de contenidos

Simplemente pulsando la opción de "Editar" se accederá a una nueva pantalla donde se podrá modificar tanto el enunciado como la solución. Una vez hechos los cambios oportunos se pulsará el botón "Guardar".

### Editar contenidos

Enunciado:	<input type="text" value="1511"/>
Solucion:	<input type="text" value="1000 + 500 + 10 + 1"/>

[Cancelar](#) [Guardar](#)

**Figura 6.7** – Ejemplo de cómo editar información

Y la información quedará actualizada.

Niveles		
ID	Contenido	Acciones
1	100 + 30 + 3	<a href="#">Editar</a> <a href="#">Borrar</a>
0	1000 + 500 + 10 + 1	<a href="#">Editar</a> <a href="#">Borrar</a>

[Volver](#)

**Figura 6.8** – Lista de contenidos actualizada



## 6.2 CONTENIDO DE LENGUA

Para incluir contenidos en esta categoría, igual que en el apartado anterior, hay que pulsar en "Añadir contenidos". En este caso solo hay que añadir una palabra y dar a "Incluir lista" para introducir todas las palabras que se pueden crear a partir de ésta. Se recomienda introducir una palabra entre 5 y 6 caracteres para su correcta visualización en el juego y facilitar la introducción de nuevos elementos para esta lista.

### Incluir Lista

Palabra:

[Incluir lista](#) [Volver](#)

**Figura 6.9** – Ejemplo de añadir contenido

Para completar la información de esta aplicación hay que acceder a "Ver contenidos" y pulsar en "Editar".

### Editar contenidos

Palabra principal	
Palabra:	<input type="text" value="casa"/>
Lista de palabras	
Palabra	Acciones
as	<a href="#">borrar</a>
Nueva palabra	
<input type="text" value="saca"/>	
<a href="#">Cancelar</a>	<a href="#">Guardar</a>

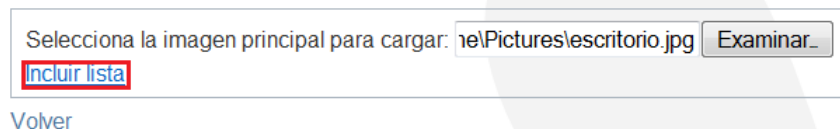
**Figura 6.10** – Ejemplo de añadir contenidos

En la parte superior se encuentra la palabra que se añadió al crear la lista. Se puede modificar en caso de haber algún error. En la parte inferior se han de añadir todas las posibles combinaciones que se puedan realizar a partir de las letras de la palabra principal. No es necesario que las palabras de la lista incluyan todas las letras de la palabra original, pueden usar solo un subconjunto de ellas.. Por eso se recomienda incluir palabras de 5 o 6 caracteres.

Por cada palabra que se incluye hay que pulsar el botón "Guardar".

### 6.3 CONTENIDO DE CIENCIAS

La aplicación de ciencias es muy similar a la de lengua con la diferencia de que hay que añadir imágenes en vez de palabras. Igual que en los otros dos apartados, se añade la información pulsando "Añadir contenido".



**Figura 6.11** – Ejemplo de añadir contenido

La imagen que se debe subir tiene que ser una imagen representativa, es decir, tiene que representar la temática de la aplicación. Por ejemplo, si la temática es geografía, esta imagen podría ser un mapa con puntos señalando los países que el alumno debe identificar, si la temática es de ciencias, podría ser un esqueleto en el que se señalan puntos que el alumno debería conocer.

El formato de la imagen tiene que ser JPG para que no existan errores en el momento de cargar esta imagen. En el caso de que la imagen no sea JPG saldrá un error por pantalla. Cuando se haya seleccionado la imagen central se pulsa en "Incluir lista".

El procedimiento a seguir es similar al que se realizaba en la temática de lengua, primero se selecciona la opción "Ver contenidos" y agregar las imágenes una por una, también en formato JPG. La idea es que cada una de estas imágenes individuales representen de forma ampliada los puntos de interés reflejados en la imagen principal. Se recomienda agregar un máximo de 5 imágenes por lista para evitar problemas de visualización. En caso de necesitar incluir más imágenes se recomienda utilizar una nueva lista para ello.

## Capítulo 7

# APLICACIÓN PARA EL ALUMNO

La aplicación para el alumno será una aplicación donde el alumno pueda realizar las actividades que normalmente realiza en el mundo real, dentro del virtual.

Para la creación de la aplicación del alumno se ha tenido en cuenta la importancia de la situación de los objetos en el mundo. Esto quiere decir que se deberá realizar una aplicación por cada rincón existente en el mundo real debido a que los alumnos de 5 y 6 años se ven influidos por la distribución y posición de los objetos. Es decir, los alumnos relacionan una actividad de aprendizaje con el lugar físico desde donde se realiza [1].

La aplicación para el alumno consistirá en tres juegos realizados en Unity3D, uno para cada temática (matemáticas, lengua y ciencias). La razón de utilizar esta plataforma va a ser por un lado, la sencillez de su programación y la posibilidad de modificar y visualizar el contenido en tiempo real; y por otro lado, el colorido y jugabilidad que añade a las aplicaciones. Las aplicaciones se podrían haber realizado en Wonderland pero se ha descartado esta opción debido a que uno de los objetivos de este proyecto era probar diferentes motores y porque el desarrollo en Unity3D es más rápido.

Una de las ventajas de Unity3D respecto a la realización de las aplicaciones es que se han pensado para que la mayor parte de código sea reutilizable. Las variaciones más importantes entre ellas será sobre todo, la forma de realizar la aplicación.

Las tres aplicaciones utilizan el mismo código para las siguientes funciones:

1. Leer archivos xml.
2. Guardar información de la actividad del usuario.

3. Detectar cuando un elemento atraviesa otro elemento.
4. Arrastrar elementos dentro del escenario.
5. Crear elementos dependiendo la información proporcionada por el profesor.

Las principales diferencias entre las aplicaciones son las siguientes:

1. Aunque todas lean archivos xml con el mismo código, no todas leen la misma información. Por ejemplo, en la aplicación de matemáticas guarda la información del xml en dos listas de igual tamaño donde una de ellas almacena los enunciados de los problemas y en otra las respuestas a dichos problemas; mientras que en las aplicaciones de lengua y ciencias almacenan una lista y una tabla del mismo tamaño, donde la lista almacena las palabras o imágenes principales y la tabla almacena un número correspondiente a cada palabra o imagen principal, y la lista de palabras o imágenes correspondiente.
2. La información de la actividad que se guarda no siempre es la misma. Por ejemplo, la aplicación de lengua guarda el número de palabras acertadas por el alumno, dato que no necesitan las otras dos aplicaciones.
3. La forma de detectar que un elemento atraviesa a otro es igual en las tres aplicaciones con la diferencia de que no tratan la información de la misma manera. En matemáticas, cuando una bola entra en un cubo, almacena el número de bolas que lo atraviesa; en lengua detecta la letra que se a colocado y en que posición; y en ciencias, detecta si la imagen que se ha colocado en la casilla en la posición correcta.

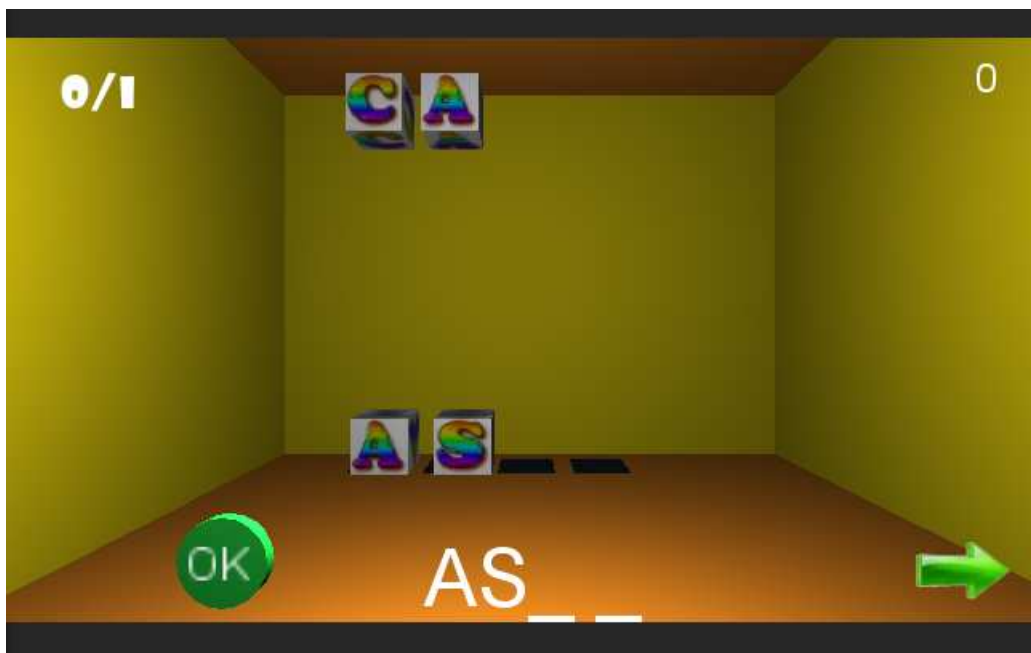
La aplicación de **Matemáticas** va a tener un escenario ya predefinido por lo que la única información adicional que necesita para que funcione la aplicación es leer el archivo xml correspondiente. En este escenario, el alumno dispondrá de cuatro cajas correspondientes a las unidades de mil, centenas, decenas y unidades y con solo pulsar el botón del ratón, se crearán bolas que deberán caer en su caja correspondiente hasta un máximo de 9 por caja. Cuando el alumno da por resuelto el ejercicio simplemente pulsará un botón para comprobar si su resultado es correcto. Si está mal, se pondrá el cielo de color rojo, por lo que tendrá que rectificar el resultado; en caso contrario, el cielo se pondrá de color azul y se pasará al siguiente nivel si lo hay. Por cada nivel que se consiga completar se guardará la información en un xml con el nombre del alumno, la nota, los aciertos, los fallos y el nivel máximo

conseguido.



**Figura 7.1** – Aplicación de Matemáticas

La aplicación de **Lengua** tendrá un escenario ya predefinido igual que la aplicación de matemáticas. Una vez extraída la palabra principal del xml, se creará un cubo por cada una de sus letras. Estos cubos se deberán arrastrar sobre unas casillas para formar palabras. Si la palabra formada coincide con cualquiera de la lista predefinida por el profesor se le sumará un acierto, y en caso contrario, un fallo. Puesto que puede ser difícil completar todas las palabras de un nivel, el usuario podrá saltar al siguiente nivel sin completar el anterior. Para ello tendrá que hacer uso de la flecha que aparece en la esquina inferior derecha 7.2. Si se completan todas las palabras del nivel, se saltará al siguiente nivel directamente en caso que exista un nivel superior. Por cada palabra que se complete se guardará la información en un archivo xml con el nombre del alumno. La información almacenada será el nombre del alumno, los fallos obtenidos y el número de palabras completadas y el nivel máximo alcanzado.



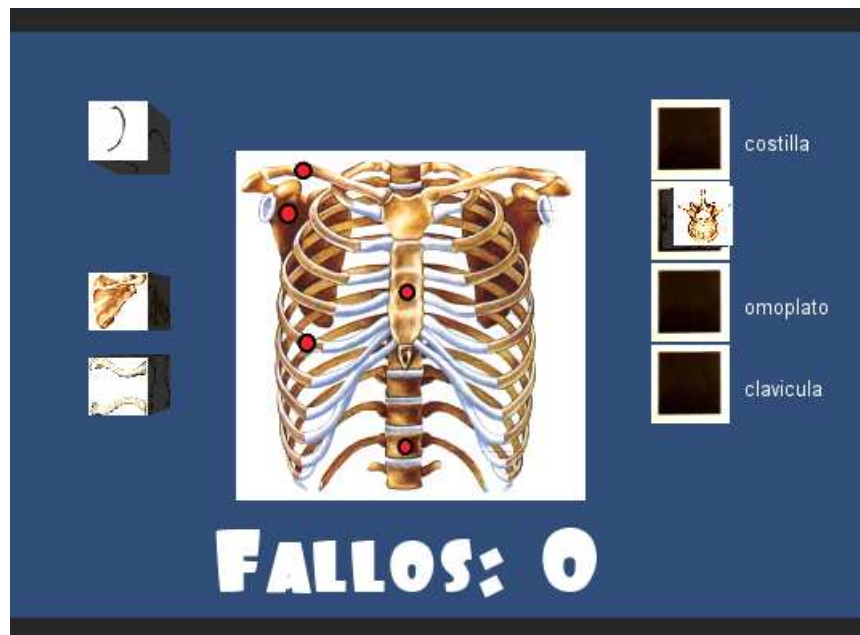
**Figura 7.2** – Aplicación de Lengua

La aplicación de **Ciencias**, a diferencia de las otras dos, no dispondrá de un escenario predefinido, sino que pintarán los elementos respecto a la información almacenada en el xml. Esta aplicación solo cargará imágenes a partir de las rutas de las imágenes. La imagen principal aparecerá en el centro de la pantalla con unos puntos de interés previamente definidos por el profesor. Las imágenes relacionadas con la imagen principal aparecerán en la parte izquierda de la pantalla. En la parte derecha aparecerán un número de casillas igual al número de imágenes y encima de estas, el nombre de cada una de las imágenes. Estas casillas aparecerán en diferente orden respecto a las imágenes. El alumno simplemente tendrá que arrastrar cada imagen a su casilla correspondiente.

El código de arrastrar las imágenes es el mismo que se utilizó en la aplicación de lengua con la diferencia de que en esta aplicación cuando se coloca una imagen encima de una casilla que no le corresponde, la aplicación responde inmediatamente si lo has hecho mal o lo has hecho bien; en cambio en la aplicación de lengua es necesario pulsar un botón para hacer la comprobación.

En esta aplicación solo se guarda la información una vez completado el nivel debido a que la resolución del mismo puede ser más rápida. Se guardará el nombre del usuario, el número de fallos obtenidos y el nivel máximo alcan-

zado.



**Figura 7.3** – Aplicación de Ciencias

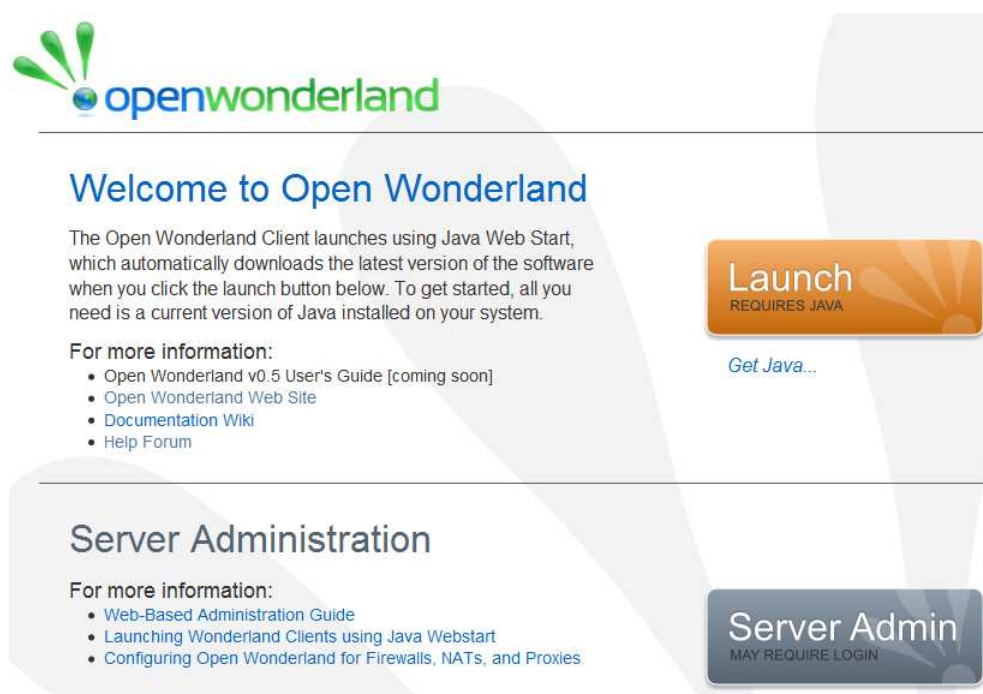
Se crearán tres directorios por cada una de las temáticas para almacenar la información de los usuarios. La información de este modo quedará mejor clasificada.

## Capítulo 8

# MANUAL DEL ALUMNO

En este apartado se explicará como acceder al mundo y utilizar las aplicaciones.

Para acceder a la aplicación solo se necesita conocer la IP de la máquina donde se ha ejecutado Wonderland. Esta IP se escribe como si fuese una URL dentro de un navegador y se accederá a una web con dos botones. Hay que pulsar el botón que pone "Launch" para iniciar la aplicación.



**Figura 8.1** – Pantalla principal

Al hacerlo aparece una ventana de "login" donde habrá que introducir un



usuario y una contraseña. Esta opción es para los usuarios que han sido previamente registrados en el sistema. Es importante no confundir la lista de usuarios registrados que pueden acceder a la aplicación con la lista de usuarios evaluables que es la que crea el profesor para identificar aquellos alumnos para los que quiera conservar un registro de actividad.



**Figura 8.2** – Ventana de login

Se elegirá la opción "Guest" para entrar en el mundo y posteriormente se pulsará "Login".

Cuando se accede se carga el mundo y aparecerá un avatar que representa al usuario y se podrá mover por todo el entorno utilizando las flechas del teclado.



**Figura 8.3** – Pantalla del cliente

Para poder utilizar las aplicaciones es necesario acercarse al edificio y en cuanto exista contacto con él, el avatar se teletransportará dentro del aula donde se encontrarán tres aplicaciones situadas en sus rincones correspondientes.



**Figura 8.4** – Rincones dentro del aula

En las tres aplicaciones aparecerá inicialmente una ventana de login. Si se dispone de usuario y contraseña el alumno podrá ser evaluado. ya que se guarda registro de su actividad. En caso contrario se puede acceder mediante la opción de "Entrar como invitado", pero en este caso no queda huella de su actividad en el sistema.



**Figura 8.5** – Login de la aplicación

Seguidamente se explicará el funcionamiento de cada una de las aplicaciones y cómo se utilizan.

## 8.1 APLICACIÓN DE MATEMÁTICAS

Una vez cargado el juego, se verá un escenario con cuatro cajas correspondientes a las unidades de mil, centenas, decenas y unidades. En la parte superior izquierda se puede ver el nivel actual y cuantos niveles hay disponibles del juego. En la parte superior derecha está situada la nota que se actualizará a medida que se avance de nivel. En la parte inferior hay un rectángulo naranja donde aparece el enunciado del problema.

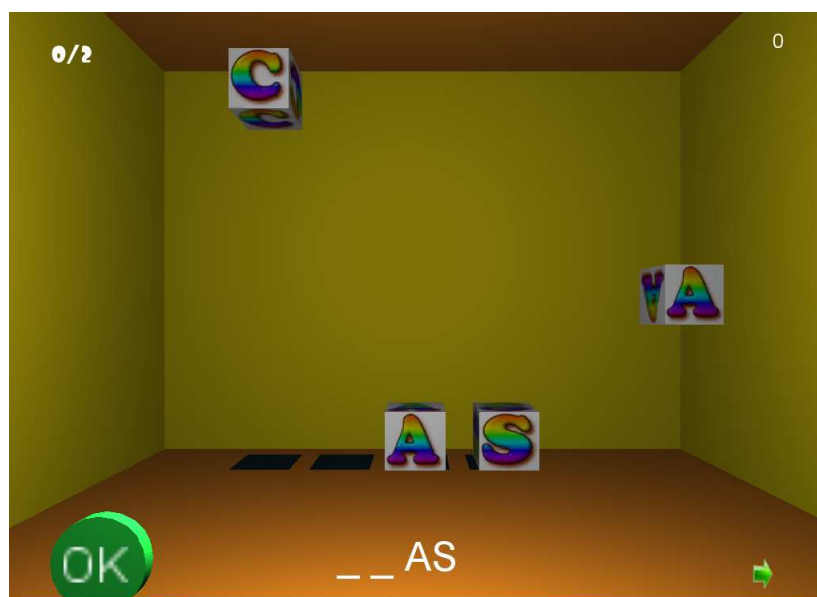


**Figura 8.6** – Juego de Matematicas

Al hacer click con el ratón por encima de las cajas, se crearán bolas rojas que caerán dentro de las cajas. El número resultante a las bolas metidas en cada una de sus cajas se verá reflejado. Si el alumno se confunde y ha introducido bolas de más en alguna de las cajas, puede corregirlo pulsando el botón derecho del ratón sobre la caja correspondiente y seleccionando con el botón izquierdo podrá borrar las bolas sobrantes. Cuando el alumno cree que el resultado es correcto deberá pulsar el botón "OK". Si el resultado es correcto se pondrá el cielo azul, se cambiará de nivel, se borrarán todas las bolas creadas, aparecerá el nuevo enunciado y se actualizarán el nivel y la nota. En caso contrario, el cielo se pondrá en color rojo y deberá corregir el resultado. Este proceso se repetirá hasta completar todos los niveles.

## 8.2 APLICACIÓN DE LENGUA

En este juego se muestra una habitación con unos cubos en la parte superior, donde cada uno tiene una letra. En la parte inferior hay unas marcas de color negro donde encajan cada uno de los cubos. En la parte superior izquierda se muestra el número de palabras acertadas y el número de palabras totales del nivel actual. Arriba a la derecha se muestra la nota, y en la parte inferior izquierda un botón "OK" para verificar las respuestas y a la derecha una flecha que sirve para cambiar de nivel aunque no se haya completado el actual.



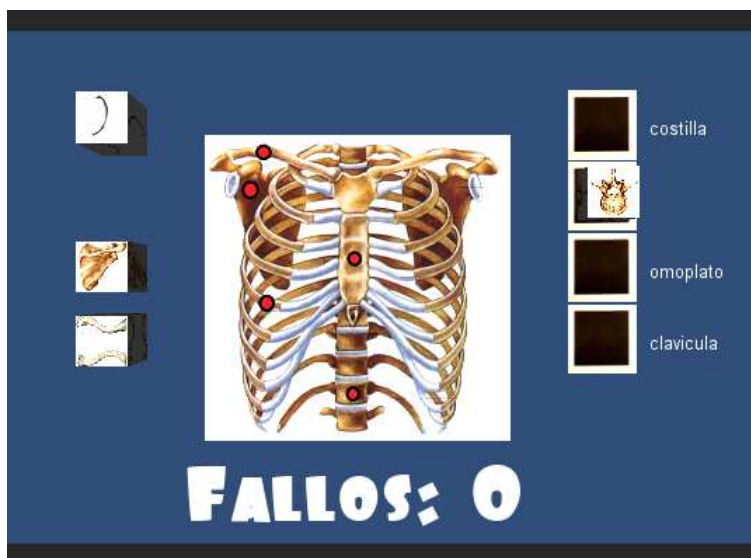
**Figura 8.7** – Juego de Lengua

El funcionamiento del juego es simple. El usuario deberá arrastrar los cubos desde la parte superior hacia la parte inferior formando palabras. La palabra formada según la colocación de los cubos se verá actualizada en la parte central inferior. Una vez construida una palabra completa habrá que pulsar "OK" para verificar. Si la palabra es correcta se actualizará el número de aciertos y los cubos volverán a su posición para que se pueda formar nuevas palabras. Si la palabra no existe porque no ha sido definida previamente, los cubos se colocarán en su posición original sin sumar ningún acierto, pero habrá una repercusión en la nota. Si la palabra es correcta pero existe, se vuelven a colocar los cubos en sus posiciones originales pero no se suma ni aciertos ni fallos. Cuando se completan todas las palabras se pasa automáticamente de nivel y así hasta completar todos los niveles definidos.

Si el alumno se atasca en un nivel y no consigue completarlo, puede saltar al siguiente pulsando la flecha pero esta acción no es reversible. Repercutirá en la nota y el profesor verá reflejado el número de palabras que el alumno ha completado.

### 8.3 APLICACIÓN DE CIENCIAS

Esta aplicación es muy semejante a la de lengua. En la parte central aparecerá una imagen principal con la que el alumno puede identificar la temática del problema a resolver. En la parte izquierda habrá unos cubos con fotografías de distintos fragmentos de la imagen central. En la parte derecha existen tantas casillas como fragmentos de imagen, cada una de ellas identificada con el nombre del elemento y colocadas en diferente orden respecto a los cubos de imagen. Por último, en la parte inferior de la pantalla, se muestra el número de fallos.



**Figura 8.8** – Juego de Ciencias

La sistemática del juego es muy parecida al juego de lengua. El usuario deberá arrastrar los cubos que contienen la imagen a sus correspondientes casillas que contienen el nombre de la imagen. A diferencia de lo que ocurría en el juego de lengua, en este caso el juego tiene respuesta inmediata. Si el alumno arrastra un cubo a una casilla que no le corresponde, el sistema sumará un fallo. Si es correcta, el cubo se quedará fijo en su casilla. Este proceso se repetirá hasta conseguir poner todos los cubos en sus casillas.

correspondientes. Una vez completado el nivel se pasará al siguiente.

## Capítulo 9

# VALIDACIÓN E INTEGRACIÓN

Completadas las aplicaciones, tanto del profesor como la del alumno, se procede a realizar pruebas de verificación e integración.

Las pruebas técnicas de verificación han sido las siguientes:

- a) Integración de la aplicación del profesor dentro de Wonderland. Se comprobó que no hubiese problemas y que funcionase todas las opciones disponibles en la aplicación.
- b) Comprobar que el archivo de configuración creado en la aplicación del profesor se pueda leer en la aplicación del alumno. Se realizaron pruebas independientes por cada una de las aplicaciones y solo quedaba comprobar que no existe problemas en la lectura y escritura de estos ficheros.

También hubo verificación por parte del usuario final, en este caso la misma profesora, que se entrevistó para la toma de requisitos.

- a) Se le presentó los modelos del edificio y del aula y fueron aceptados sin ningún problema. Echaba de menos la decoración del aula que sería posterior a la prueba de las aplicaciones.
- b) Los tipos de juego que se mostraron fueron ejemplos de posibles escenarios. Se les mostró un juego donde se mostraba una isla donde uno podía moverse por ella y un juego en tercera persona donde el personaje era un gusano que se movía por la pantalla. La profesora comentó verbalmente el tipo de actividades que realizan los alumnos y quedó en enviar la información que debía tener el juego (imágenes, audios...). Puesto que esta información no llegó, se realizaron juegos acorde al tipo de ejercicios que comentó verbalmente.

c) Quedaron dos verificaciones pendientes:

1. Validación del prototipo final.
2. Prueba con alumnos.

Estas dos últimas verificaciones no se han podido realizar porque requieren permiso por parte de los padres de los alumnos.

Por otro lado, se ha procedido a la integración de las aplicaciones realizadas en Unity3D con Wonderland en Linux y Windows. En Windows no es posible la integración porque no está habilitada la opción de agregar aplicaciones externas así que finalmente se optó por hacer las pruebas de integración en Linux. Los problemas encontrados han sido los siguientes:

- Imposibilidad de incluir las aplicaciones de Unity3D dentro de Wonderland. Esto ha ocurrido porque las aplicaciones de Unity3D están hechas para Windows.
- Imposibilidad de incluir las aplicaciones de Unity3D como una aplicación web en Wonderland. La página web carga pero necesita un plugin del navegador (archivo .exe)
- El problema de incluir la aplicación de Unity3D como una página web, es que impide el acceso a documentos disponibles del ordenador. Esto podría solventarse con la versión Pro de Unity3D, que requiere licencia, pero ya que el proyecto no tiene financiación no ha sido viable.
- Alta carga computacional. Debido a los altos requisitos que necesita Wonderland se le suman los que pueda necesitar Unity3D y el resultado es que la aplicación va mucho más lenta.
- Incertidumbre sobre el futuro de Wonderland. Debido a la decisión tomada por Oracle de discontinuar el soporte de la plataforma, esta tiene un futuro incierto.

Debido a los anteriores inconvenientes, se ha realizado la incorporación total de toda la aplicación en Unity3D. Esta integración supone las siguientes ventajas respecto a la integración con Wonderland:

1. Da la funcionalidad que se necesita. Los modelos de Google Sketchup son totalmente integrables en Unity3D por lo que se puede recrear el escenario con las aplicaciones incluidas.



2. Es más rápido en la carga y más sencillo de instalar.

Esta solución aunque es buena, también se han encontrado sus inconvenientes a la hora de realizarla:

1. Unity3D no es un mundo virtual, por lo que la funcionalidad para que los usuarios se conecten desde otro ordenador y mantener la actividad del usuario no se consigue.
2. Si se incluye como un navegador, todos los usuarios podrían conectarse desde un navegador instalando un plugin. El problema que se requeriría la versión Pro de Unity3D.

La integración de estas dos plataformas estará pendiente como trabajo futuro.

## Capítulo 10

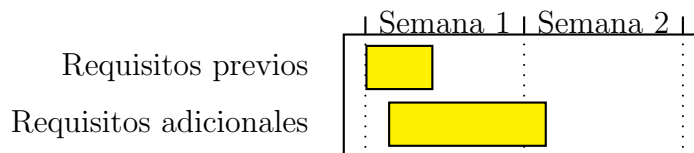
# PLAN DE TRABAJO Y PRESUPUESTO

### 10.1 PLAN DE TRABAJO

El proyecto transcurre desde Marzo de 2010 hasta Noviembre de 2010. Se mostrará la planificación del proyecto y el tiempo dedicado a cada una de las actividades que lo compone.

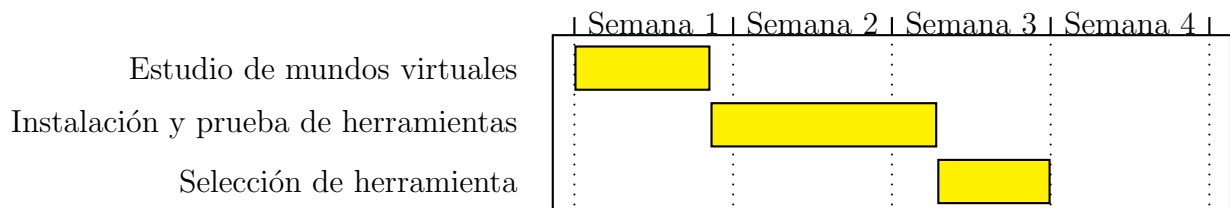
A continuación se muestra el diagrama Gantt de cada uno de los apartados de los que se compone este proyecto.

#### 10.1.1 REQUISITOS INICIALES



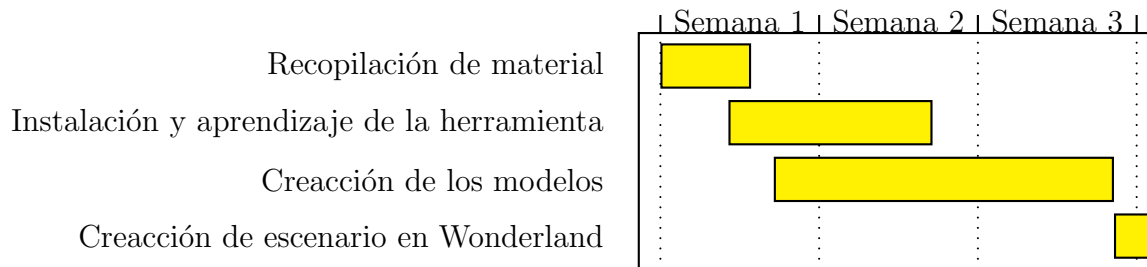
Desde el 1 de Marzo hasta el 10 de Marzo.

#### 10.1.2 ESTADO DEL ARTE



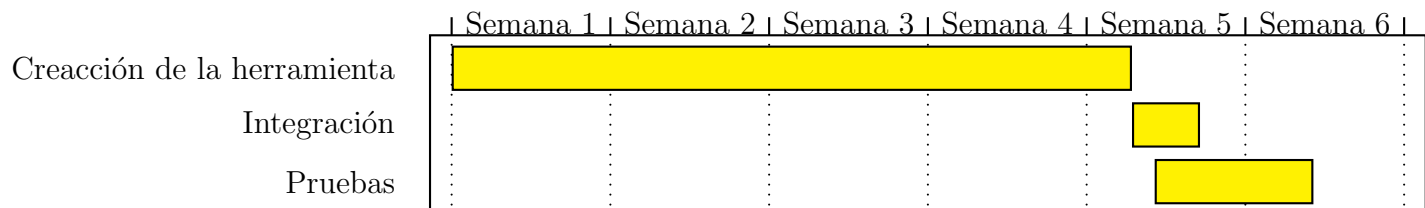
Desde el 10 de Marzo hasta el 31 de Marzo.

### 10.1.3 CREACIÓN DE MUNDOS 3D (SKETCHUP) Y CREACIÓN DE ESCENARIO EN WONDERLAND



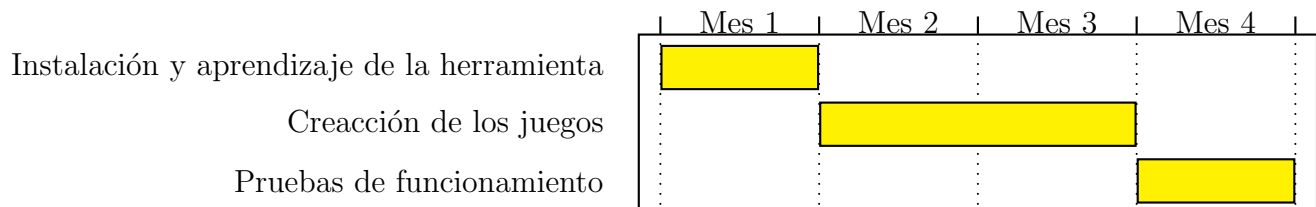
Desde el 1 de Abril hasta el 30 de Abril.

### 10.1.4 HERRAMIENTA DEL PROFESOR Y PRUEBAS



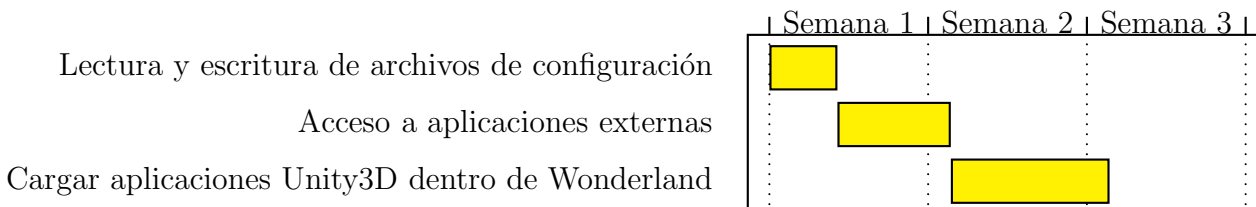
Desde el 1 de Mayo hasta el 9 de Junio.

### 10.1.5 CREACIÓN DE JUEGOS EN UNITY3D Y PRUEBAS



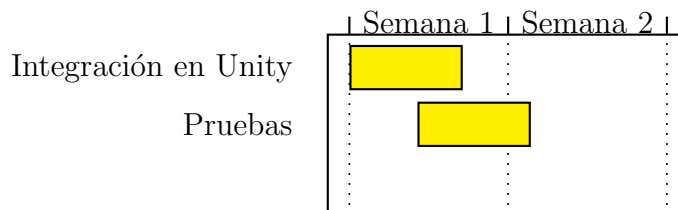
Desde el 9 de Junio hasta el 9 de Octubre.

### 10.1.6 PRUEBAS DE INTEGRACIÓN



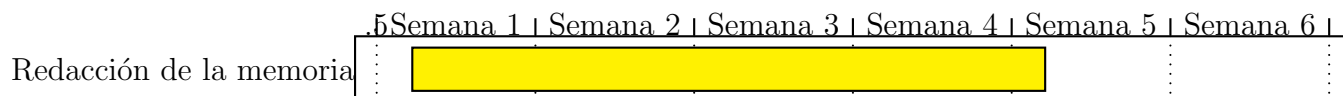
Desde el 9 de Octubre hasta el 21 de Octubre.

### 10.1.7 INTEGRACIÓN FINAL



Desde el 21 de Octubre hasta el 2 de Noviembre.

### 10.1.8 REDACCIÓN DE LA MEMORIA



Desde el 2 de Octubre hasta el 15 de Noviembre.

## 10.2 PRESUPUESTO

En esta sección se mostrará el presupuesto calculado para el proyecto. Incluye todos los gastos de desarrollo y puesta en funcionamiento.

El tiempo requerido para la realización de este proyecto ha sido de 8 meses y 15 días.

El trabajo se ha realizado a razón de 5 horas diarias, durante el periodo de 8 meses suponiendo 20 días hábiles de media al mes. Esto supone un total de 850 horas.

### 10.2.1 DESGLOSE POR ACTIVIDADES

En la tabla siguiente se muestra las actividades que se han llevado a cabo para la realización del proyecto con la duración de cada una de ellas.

Tarea	Horas
Análisis	125 horas
Diseño	540 horas
Integración y pruebas	60 horas
Documentación	125 horas

**Tabla 10.1** – Actividades y duración de las mismas

### 10.2.2 SALARIOS POR CATEGORÍA

Para la elaboración del proyecto se requiere personal cualificado para cada una de las actividades.

Para realizar los cálculos se han realizado las siguientes suposiciones:

- La jornada laboral es de 5 horas.
- 20 días laborables al mes.
- El número de horas del proyecto es de 850 horas.

Cargo	Horas	Coste/Hora	Sueldo Bruto/mes
Analista	125 horas	12,5 €	2.000 €
Responsable de diseño	540 horas	12,5 €	2.000 €
Responsable de integración y pruebas	60 horas	12,5€	2.000 €
Responsable de documentación	125 horas	9,375€	1.500 €

**Tabla 10.2** – Tabla de salarios

### 10.2.3 GASTOS DE PERSONAL

Este proyecto ha sido realizado por una sola persona, asumirá todos los roles del apartado anterior para desarrollar cada una de las actividades que forman el proyecto. El importe total es de 7.500€.

### 10.2.4 RECURSOS MATERIALES EMPLEADOS

En la tabla 10.3 continuación mostrará los materiales necesarios para la implantación del proyecto. Hay que destacar que el coste del software es cero debido a que el software que se ha utilizado es libre.

Los materiales incluyen el I.V.A y el total de todos los recursos asciende a 453.15€.

## 10.3 COSTES INDIRECTOS

Como costes indirectos se aplicará un 10% del total.  
 $(7500 + 453,15) * 0,1 = 795,315\text{€}$ .

Recurso	Cantidad	Precio Unitario	Coste total
Placa base Gigabyte GA-P55-US3L	1	83€	83€
Procesador Intel Core i3 540 3.06Ghz Box Socket 1156	1	100€	100€
Disco Duro Seagate Barracuda 7200.12 500GB SATA2	1	37.95€	37.95€
Memoria RAM Kingston ValueRAM 4GB DDR2 800 PC2-6400	2	41.5€	83€
Tarjeta gráfica Zotac GeForce GTS 250 1GB GDDR3	1	107€	107€
Carcasa Ordenador B-Move Eclipse Silver	1	24.95€	24.95€
Ventilador Thermaltake Slim X3 CPU Cooler	1	17.25€	17.25€
Software utilizado	-	0€	0€

**Tabla 10.3** – Tabla de recursos necesarios

## 10.4 MARGEN DE BENEFICIO

Se aplicará un margen de beneficios sobre el 15% del total del coste del proyecto.  $(7500 + 453,15 + 795,315) * 0,15 = 1312,26975€$ .

## 10.5 RESUMEN DEL PRESUPUESTO

El presupuesto total del proyecto es de 10060,73475€.

Recurso	Coste total
Personal al cargo	7500€
Recursos materiales	453,15€
Costes indirectos	795,315€
Beneficio	1312,26975€
<b>TOTAL</b>	<b>10060,73475€</b>

**Tabla 10.4** – Presupuesto total

## Capítulo 11

# CONCLUSIONES

En este apartado se comentarán las conclusiones obtenidas tras la realización de este proyecto.

Durante su realización se han tenido que resolver problemas a distintos niveles, desde la toma de requisitos y validación con el usuario final, pasando por cuestiones de diseño gráfico, hasta la implementación haciendo uso de diferentes motores gráficos y sus implementaciones para distintos sistemas operativos. Se ha trabajado principalmente con dos herramientas: Wonderland, una plataforma para la creación de mundos virtuales y Unity3D, un motor de juegos 3D.

Wonderland es una plataforma que necesita grandes requisitos gráficos. Tiene varias limitaciones debido a que no funciona igual en diferentes plataformas. Debido a que las funciones necesarias para insertar aplicaciones externas dentro de la plataforma Wonderland solo podía funcionar en Linux, se decidió utilizar este sistema operativo.

Surgieron problemas a la hora de la integración de las aplicaciones creadas con Unity3D dentro de Wonderland, debido a que la aplicación resultante que devuelve Unity3D es un archivo .exe, ya que este formato no es legible por Linux.

Unity3D es una plataforma libre desde Enero de 2010 que poco a poco ha ido integrándose en más sistemas operativos y en un futuro posiblemente aparezca Unity3D para Linux. Esto facilitaría mucho la integración de estas dos plataformas, pero de momento quedará como trabajo futuro la integración de Unity3D con Wonderland.

También se ha visto que la jugabilidad de una aplicación realizada con

Unity3D requiere menos carga del procesador y mejora considerablemente los gráficos comparado con Wonderland, por lo que hace pensar que sería mejor realizar el mundo completamente en Unity3D, algo complejo porque sería crear la arquitectura cliente-servidor desde cero.

Pese a todos los problemas encontrados se ha conseguido cumplir el objetivo principal de conseguir una aplicación formada por un mundo 3D creado a imagen y semejanza de un colegio real e integrar en él un conjunto de juegos que replicasen algunas de las actividades básicas que el alumno hace en el aula.

Por último se ha diseñado una aplicación para el profesor sencilla de manejar que mediante un formulario permite cambiar el contenido de los ejercicios sin necesidad de conocimientos avanzados, proporcionando únicamente texto y la ubicación de ficheros de imágenes que forman parte del ejercicio.



## Capítulo 12

# TRABAJO FUTURO

Las posibles vías que abre este proyecto son las siguientes:

1. Creación de grupos. Este proyecto está pensado para un aula y un grupo de alumnos en concreto, por lo que la creación de grupos posibilitaría a otro grupo a utilizar la aplicación y cada uno tendría sus actividades.
2. Prueba en un entorno real. Realizando esta prueba se podría comprobar si realmente los alumnos se sienten identificados con el entorno y manejan la aplicación como si se tratase del mundo real.
3. Integración de Unity3D en Wonderland. Sería necesario adquirir una licencia de Unity Pro para poder resolver por completo los problemas de integración.
4. Ampliación de temáticas, como por ejemplo, inglés, música o plástica.
5. Incorporación de otro tipo de archivos por parte del profesor. Las aplicaciones creadas tratan archivos de texto e imágenes, por lo que sería interesante tratar otro tipo de archivos, como por ejemplo audios, para una aplicación de música.

## Capítulo 13

# ANEXO 1: MANUAL DE INSTALACIÓN

### 13.1 INSTALACIÓN DE WONDERLAND

En esta sección se explicarán los pasos necesarios para la instalación de Wonderland en un ordenador.

Este proceso de instalación se ha probado en los siguientes equipos:

- Ordenador portatil HP con sistema operativo Windows 7 y microprocesador AMD Turion(tm) X2 Ultra Dual-Core Mobile ZM-82 2.20 GHz, con 3GB de memoria RAM y tarjeta gráfica ATI Radeon HD 3200 Graphics.
- Ordenador portatil Dell con Windows 7 y microprocesador Intel Core Duo CPU 2.53Ghz, con 8GB de memoria RAM y tarjeta gráfica Mobile Intel(R) 4 Series.
- Ordenador de sobremesa con sistema operativo Ubuntu y microprocesador AMD Athlon 64 x2 4200+, con 1GB de memoria RAM y tarjeta gráfica nVidia GT400.

La instalación se ha probado sobre el sistema operativo Windows y Linux. Se recomienda la instalación sobre Linux porque sobre Windows se han encontrado las siguientes limitaciones:

- Imposibilidad de incluir aplicaciones externas dentro de Wonderland.
- Imposibilidad de que otros usuarios accedan a Wonderland desde otra máquina.

En cualquier caso, la instalación de Wonderland sería exactamente igual para ambos sistemas operativos.

1. En primer lugar se debe comprobar que se tiene instalado JAVA en el sistema. Si se está utilizando Linux hay que abrir un terminal y escribir el siguiente comando:

```
> /usr/jdk/bin/java -version
```

Si se tiene instalado aparecerá algo similar a lo siguiente, donde "nn" indica el número de versión y puede ser cualquier número:

```
java version "1.6.0_nn" Java(TM) SE Runtime Environment (build 1.6.0_16-b01)
Java HotSpot(TM) Server VM (build 14.2-b01, mixed mode)
```

En Windows hay que entrar en Mi PC e ir a C:\\Archivos de Programa\\y comprobar que existe un directorio llamado Java. En caso afirmativo acceder a ella y ver si hay una carpeta llamada jdk1.6.0\_nn.

Hay que comprobar que se tiene al menos la versión 1.6.0\_nn. De todos modos se recomienda tener la última versión disponible que se puede descargar desde esta web:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html> eligiendo la opción JDK 6 Update XX (XX = versión).

2. A continuación hay que descargarse la aplicación ANT [5] y SUBVERSION [17]. ANT se utilizará para compilar y ejecutar Wonderland y SUBVERSION para descargar la plataforma del repositorio.

3. Una vez instaladas las dos aplicaciones anteriores, se descarga la plataforma Wonderland desde su página web

<http://code.google.com/p/openwonderland/wiki/DownloadBuildSource05>. Para ello se abre un terminal y creamos un directorio (si se quiere) con el nombre que se desee:

```
> mkdir nombre_del_directorio
```

Se accede a la carpeta:

```
> cd nombre_del_directorio
```

Y se escribe el siguiente comando para descargar la compilación de Wonderland.

```
> svn checkout https://openwonderland.googlecode.com/svn/branches/0.5-preview4
```

4. Una vez descargado, se comprueba que existe un archivo llamado build.xml. Sino es así se volvería a repetir el comando anterior para que termine de descargar ya puede ocurrir que haya interrupciones no deseadas en la conexión. Una vez comprobado que está todo correcto hay que compilarlo con el siguiente comando:

```
> ant
```

Una vez compilado se puede ejecutar la plataforma de la siguiente manera:

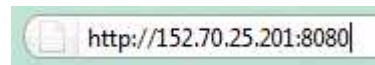
```
> ant run-server
```

6. Empezará a salir información por pantalla y finalmente se detendrá el flujo de información y deberá aparecer lo siguiente:

```
[java] at org.jdesktop.wonderland.runner.BaseRunner.start(BaseRunner.java:220)
[java] ... 12 more
[java] -----
[java] Wonderland web server started successfully.
[java] Log files are in          /.wonderland-server/0.5-preview4/log
[java] Web server running on http://152.70.25.201:8080/
[java] -----
```

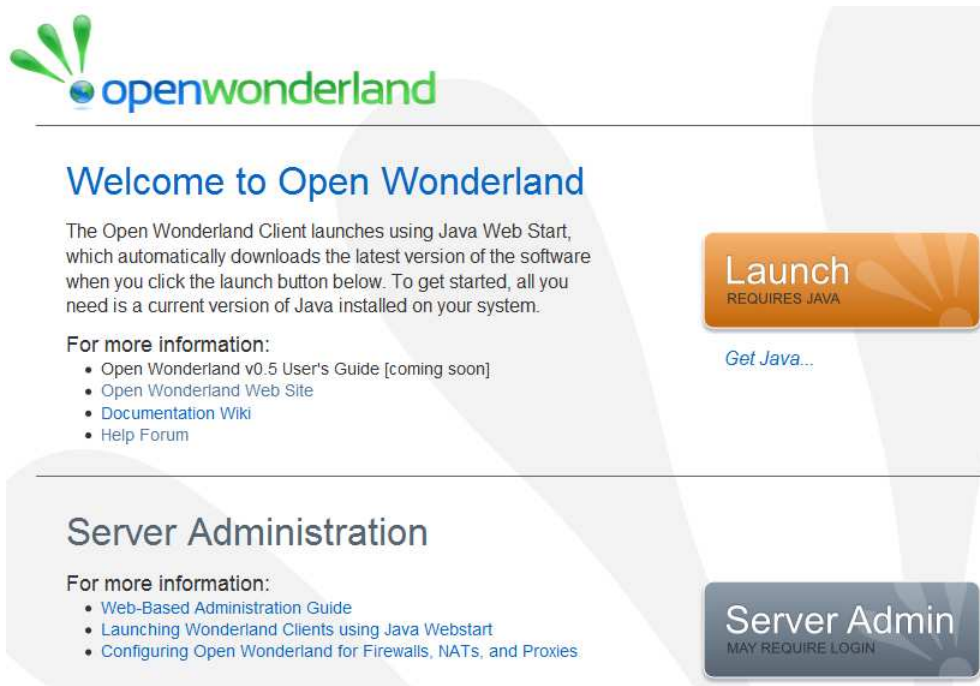
**Figura 13.1** – URL de Wonderland

La IP 152.70.25.201:8080, en este caso será la IP del servidor con la que se podrá acceder a Wonderland. Para acceder, se copia la IP, se abre una ventana de Firefox y la se pega como si fuese una URL. Se puede utilizar también otro navegador pero puede haber problemas con la visualización de los contenidos del servidor. Si se utiliza otro navegador diferente a Firefox puede ocurrir que la información del servidor no se visualice correctamente.



**Figura 13.2** – Acceso a Wonderland desde un navegador

Aparecerá una ventana como la siguiente:



**Figura 13.3** – Pantalla principal

El botón superior permitirá acceder a la parte cliente de Wonderland y el inferior a la parte servidor.

Otra opción para instalar y ejecutar Wonderland sería descargar la plataforma comprimida en un archivo jar, normalmente llamado Wonderland.jar y ejecutarlo. Para ello habría que descargarse la plataforma desde esta otra URL: [http://openwonderland.org/modules/module-warehouse/doc\\_download/220-05-preview-4](http://openwonderland.org/modules/module-warehouse/doc_download/220-05-preview-4)

Una vez descargado el archivo, hay que abrir un terminal, colocarse en la carpeta que contiene el archivo y ejecutar el siguiente comando:

```
> java -jar Wonderland.jar
```

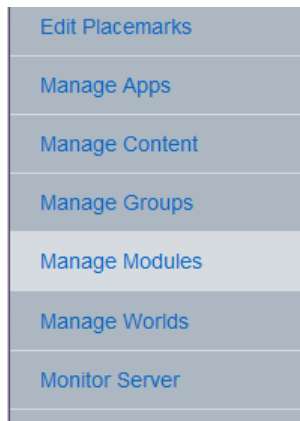
Seguidamente se repetirá el proceso anterior desde el paso 6.

## Configuración de Wonderland

Inicialmente el acceso a la parte servidor no presenta ningún tipo de restricción y esto puede suponer un problema de seguridad. Para solucionarlo se

tiene que instalar un módulo de seguridad, que consiste en un archivo .jar donde se puede indicar las personas con privilegios de acceso. [12] A continuación se describe el proceso y se puede encontrar información adicional en la web del proyecto. [13]

Hay que entrar en la parte servidor y seleccionar la casilla "Manage Modules".



**Figura 13.4** – Opción "Manage Modules" del menú

Se selecciona en la lista de módulos uno llamado "security-session-noauth". A continuación hay que pulsar en "Remove selected modules".

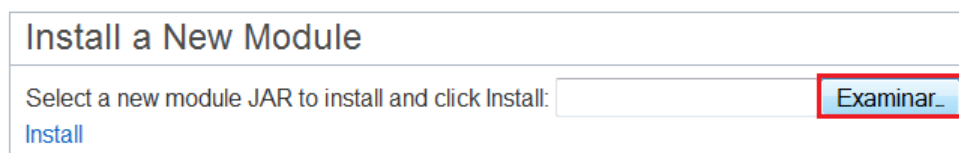
<input checked="" type="checkbox"/>	security-session-noauth	v0.5	Security no authorization session store library
<input type="checkbox"/>	servermanager	v0.5	Server manager library
<input type="checkbox"/>	sharedstate	v0.5	Shared state API and library
<input type="checkbox"/>	snapshot-manager	v0.5	Snapshot manager library
<input type="checkbox"/>	textchat	v0.5	Text chat API and library
<input type="checkbox"/>	viewproperties	v0.5	View properties configuration library
<input type="checkbox"/>	voicebridge	v0.5	jVoicebridge server
<input type="checkbox"/>	webdav	v0.5.1	Webdav library
<input type="checkbox"/>	xapps-config	v0.5	Shared apps configuration library
<input type="checkbox"/>	xremwin	v0.5	X11 shared application library

[Removed selected modules](#)

**Figura 13.5** – Eliminar un módulo

Aparecerá una página de confirmación donde se pulsará "OK".

A continuación, dentro de "Manage Modules" se pulsa en el botón examinar.



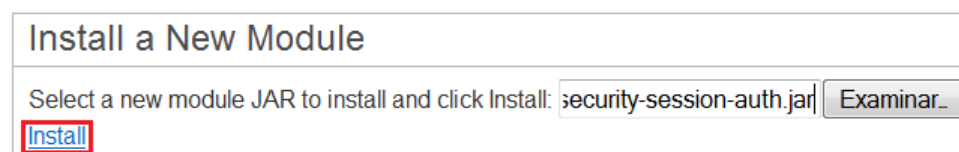
**Figura 13.6** – Instalación de un módulo

Se abrirá una nueva ventana donde hay que seleccionar el módulo.



**Figura 13.7** – Selección de un módulo

Una vez seleccionado el archivo se procede a instalarlo pulsando a "Install".



**Figura 13.8** – Proceder a instalar

Si el módulo se ha instalado correctamente aparecerá el siguiente mensaje:

### Module Install Success

The module installation was successful. The Darkstar server may need to be restarted to complete the installation. Please consult the Pending Modules table for the list of modules requiring a Darkstar server restart.

OK

**Figura 13.9** – Confirmación de la instalación

Hay que hacer pulsar en "OK" para que se reinicie el servidor y se apliquen los cambios.

A continuación, una vez reiniciado el servidor aparecerá la siguiente pantalla:

## Log in

Username:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Log In"/>	

**Figura 13.10** – Pantalla de login

Hay que introducir un usuario y una contraseña con la cual se podrá acceder a la parte servidor de Wonderland. Por defecto el usuario y la contraseña es "admin". Una vez se ha accedido al servidor se podrá ver una nueva opción en el menú llamada "Manage Users" que hay que seleccionar.



**Figura 13.11** – Nueva casilla del menú

De los cuatro usuarios que aparecen por defecto, es necesario modificar al menos la contraseña que viene por defecto en "admin" para poner una nueva que proteja el sistema..

## View Users

Users			
Name	Id	Email	Actions
System Administrator	admin		<a href="#">edit</a> <a href="#">remove</a>
Darkstar Server	darkstar		<a href="#">edit</a> <a href="#">remove</a>
Web Server	webserver		<a href="#">edit</a> <a href="#">remove</a>
Shared App Server	sasxprovider		<a href="#">edit</a> <a href="#">remove</a>

[Add user](#)

**Figura 13.12** – Usuarios por defecto en Wonderland

Para ello se presiona "edit" y se procede a realizar el cambio.



## Details for user admin

User Id:	<input type="text" value="admin"/>
Full Name:	<input type="text" value="System Administrator"/>
Email:	<input type="text"/>
Password:	<input type="password" value="•••••••"/>
Confirm Password:	<input type="password"/>
<a href="#">Cancel</a> <a href="#">Update user</a>	

**Figura 13.13** – Pantalla de edición de la información del usuario

Cuando se han terminado de introducir los datos se pulsa "Update User".

Se pueden añadir todos los usuarios que se deseen con sus respectivos nombres y contraseñas y todos ellos puedan acceder al servidor.

### 13.1.1 ACCESO A WONDERLAND DESDE OTROS EQUIPOS (OPCIONAL)

Si se desea de acceder a Wonderland desde un ordenador distinto a aquel en el que se realizó la instalación, puede haber problemas en la carga del cliente. Esto se debe a la configuración por defecto y para arreglarlo se puede modificar un fichero de configuración llamado "my.properties.file.example" disponible en el directorio de Wonderland. Con un editor de textos planto, una vez abierto el archivo, hay que descomentar la línea donde pone "wonderland.webserver.host" quitando la almohadilla (#) del principio. Justo después del igual hay que añadir la IP de la máquina donde se va a ejecutar Wonderland. Quedaría algo así: wonderland.webserver.host=xxx.xxx.xxx.xxx.

Una vez realizado este cambio, cuando se quiera ejecutar Wonderland habrá que modificar el comando de ejecución de la siguiente manera:

```
> ant run-server -Drun.properties.file=my.properties.file.example
```

Si se ha instalado el módulo de seguridad, la pantalla inicial del cliente se verá afectada y la pantalla de login solo dejará acceder a los usuarios registrados. Para dejar acceder a cualquier usuario en el mundo hay que añadir la siguiente línea en el archivo de propiedades y luego arrancar el servidor con el comando anterior:

```
wonderland.security.allow.guest.login=true
```

## 13.2 INSTALACIÓN DE LA APLICACIÓN DEL PROFESOR

Existen dos procedimientos para instalar esta aplicación denominada (addcontents).

La primera es instalarlo siguiendo el mismo procedimiento que se usó para el módulo de seguridad desde el servidor. El archivo .jar se encuentra disponible dentro del directorio \dist del módulo.

La segunda opción sería copiar el directorio del módulo directamente en la plataforma Wonderland (recomendado si se ha descargado la plataforma utilizando ANT). El directorio habrá que copiarlo en \modules\tools\. Una vez hecho esto, en este mismo directorio existe un fichero llamado build.xml que se tiene que abrir con un editor de texto cualquiera y buscar la línea:

```
<filelist dir="\${current.dir}\" id="all.projects">
```

Justo debajo hay que añadir la siguiente línea:

```
<file name="addcontents/build.xml"/>
```

De esta manera, al arrancar Wonderland el módulo se autoinstalará.

Una vez realizado este paso, si se accede al servidor, aparece una nueva opción llamada "Contenidos".

## 13.3 CARGAR MODELOS DE GOOGLE SKETCHUP EN WONDERLAND

Para cargar modelos de Google Sketchup dentro de Wonderland, tienen que estar en un formato .kmz o .dae. Lo primero que hay que hacer, una vez creado el modelo dentro de Wonderland es exportarlo.

Seleccionaremos

Archivo\Exportar\Modelo 3D...

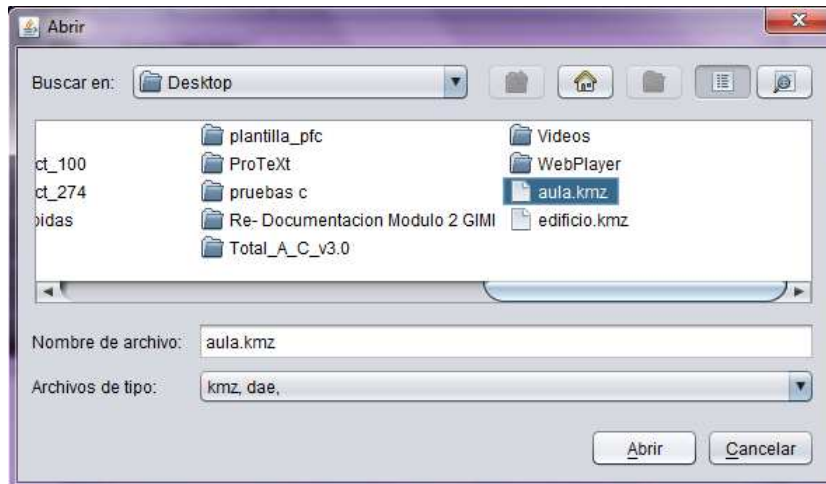
Y se abrirá una ventana que permitirá elegir el destino del archivo que se guardará en formato .kmz o en .dae.

Cuando ya se tiene el archivo hay que acceder a la parte cliente de Wonderland y una vez que este cargue, seleccionar Importar\Modelo.... y aparecerá una nueva ventana.



**Figura 13.14** – Ventana para cargar el módulo

Pulsando en el botón "Cargar modelo...", aparecerá una nueva ventana de selección para escoger el modelo que se quiera importar.



**Figura 13.15** – Ventana para seleccionar el modelo

Una vez seleccionado el modelo se pulsa en "Abrir" y tras las ventanas que informan del proceso de carga aparecerá el modelo tal y como se indica en la figura 13.16



**Figura 13.16** – Resultado final al cargar un modelo en Wonderland

# Bibliography

- [1] La pedagogía de la responsabilidad y la autoformación.  
[http://uhu.es/cine.educacion/figuraspedagogia/0\\_montessori.htm](http://uhu.es/cine.educacion/figuraspedagogia/0_montessori.htm).
- [2] Secondlife. <http://secondlife.com>, Octubre 2003 - 2010.
- [3] Open wonderland. <http://www.openwonderland.org>, 2006 - 2010.
- [4] Tecnologías xml. <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>, Enero 2008.
- [5] Ant. <http://ant.apache.org>, Noviembre 2010.
- [6] Google earth. <http://earth.google.es>, 2010.
- [7] Google sketchup. <http://sketchup.google.com>, 2010.
- [8] Google wharehouse. <http://sketchup.google.com/3dwarehouse>, 2010.
- [9] Java servlet. [http://es.wikipedia.org/wiki/Java\\_Servlet](http://es.wikipedia.org/wiki/Java_Servlet), Noviembre 2010.
- [10] Jsp. [http://es.wikipedia.org/wiki/JavaServer\\_Pages](http://es.wikipedia.org/wiki/JavaServer_Pages), Noviembre 2010.
- [11] Netbeans. <http://netbeans.org>, 2010.
- [12] Open wonderland: Authentication module.  
[http://openwonderland.org/modules/module-warehouse/doc\\_details/190-authentication?cat=add\\_ons&lang=en](http://openwonderland.org/modules/module-warehouse/doc_details/190-authentication?cat=add_ons&lang=en), 2010.
- [13] Open wonderland: Configuring authentication.  
<http://wiki.java.net/bin/view/Javadesktop/ProjectWonderlandAuthentication05>, 2010.
- [14] Open wonderland: Source. <http://code.google.com/p/openwonderland/wiki/DownloadBuildSou>, 2010.

- [15] Open wonderland: Tutorial part 1.  
[https://wonderland.dev.java.net/wiki/beginner/Wonderland\\_Tutorial\\_01.pdf](https://wonderland.dev.java.net/wiki/beginner/Wonderland_Tutorial_01.pdf),  
2010.
- [16] Opensim. [http://opensimulator.org/wiki/Main\\_Page](http://opensimulator.org/wiki/Main_Page), Septiembre 2010.
- [17] Subversion. <http://subversion.apache.org>, 2010.
- [18] Unity3d. <http://unity3d.com>, 2010.